MEMOIRE DE PROJET DE FIN D'ETUDES

Pour l'obtention du diplôme d'Ingénieur d'Etat
En
Génie Réseaux et Systèmes

Etude et mise en œuvre du service pilote ToIP de RENATER

Réalisé à : Groupement d'Intérêt Public Réseau National de Télécommunications pour la Technologie, l'Enseignement et la Recherche

Réalisé par

Mr Mohamed El Mahdi BOUMEZZOUGH

Soutenu le : 4 février 2009 devant le Jury :

Mme. R.ALASSALI	Professeur à l'ENSA de Marrakech (Présidente)
Mr. S.MUYAL	Ingénieur équipe SIPA (services IP Avancés et prospective) (Encadrant)
Mr. B.TUY	Responsable de l'équipe SIPA (Encadrant)
Mr. N.IDBOUFKER	Professeur à l'ENSA de Marrakech (Encadrant)
Mr. L.GOUJDAMI	Professeur à l'ENSA de Marrakech (Examinateur)

Année 2008/2009

Remerciements

Au terme de ce travail, je tiens à exprimer ma profonde gratitude et mes sincères remerciements à mes tuteurs de stage au GIP RENATER M. Simon MUYAL et M. Bernard TUY pour tout le temps qu'ils m'ont consacré, leur directives précieuses, et pour la qualité de leur suivi durant toute la période de mon stage.

Je tiens aussi à remercier vivement le directeur du GIP RENATER, M. Dany Vandromme qui a accepté de m'accueillir en stage au sein de son organisme.

Je voudrai remercier également tout le personnel du GIP RENATER pour sa gentillesse et son soutien notamment Mme Emilie Camisard.

Mes profonds remerciements vont à mon encadrant à l'ENSA M. Noureddine IDBOUFKER qui a accepté d'encadrer mes travaux durant ces 4 mois de stage.

Mes plus vifs remerciements s'adressent aussi à tout le cadre professoral et administratif de l'ENSA de Marrakech.

Mes remerciements vont enfin à toute personne qui a contribué de près ou de loin à l'élaboration de ce travail.

ملخص

عرفت تكنولوجيا الاتصال عبر شبكة لايبي (IP) تطورا مهمـــا، وبدأت تعرف استعمالا متزايدا في كثير من المؤسسات. تعتمد هذه التكنولوجيا على انجاز المكالمات الهاتفية عبر شبكة لايبي. ومن بين المؤسسات التي بدأت تستعمل هذه التكنولوجيا نجد الجامعات و مراكز البحث الفرنسية.

وحتى تتمكن هذه المؤسسات المتصلة بالشبكة الوطنية لـلا تصالات من اجل التكنولوجيا و التعليم و البحث العلمي من الاتصال فيما بينها عبر هذه التكنولوجيا،تم وضع نموذج تجربي لهذه الخدمة.ويعتمد هذا النموذج على خادم يعمل على توجيه المكالمات الهاتفية بين هذه المؤسسات وذلك باستعمال البرتكولSIP .

بعد التأكد من عمل هذا النموذج، كانت المرحلة الثانية هي تطوير هذه الخدمة وجعلها خدمة رئيسية لمؤسسات الشبكة الأكاديمية الفرنسية.في هذا السياق يدخل مشروعي النهـائي للدراســة.

الاهداف التي سطرت خلال هذا المشروع هي :

- **تطوير النموذج التجربي**
- **دراسة وانشاء نظام لمراقبة خدمة الاتصال عبر شبكة لايبي**
- **دراسة وانشاء نظام لاحصاء المكالمــات الهــاتفية**
- **حمــاية الخـــــادم الرئيســـــي**

وبهذا يكون هذا التقريــر نتــاج 4 اشهــر من العمل داخـل فريق خـدمــات لايبي المتطورة التابع لمؤسسة جيب غونت التي تسهر على تسييرالشبكة الوطنية لـلا تصالات من اجل التكنولوجيا و التعليم و البحث العلمي.

كلمات مفاتيح: تكنولوجيا الاتصال عبر شبكة لايبي ، SIP ، نظام مراقبة، نظام احصاء، حمــاية

Résumé

La téléphonie sur IP (ToIP) est une technologie qui s'impose progressivement dans tous les secteurs, elle consiste à faire transiter les communications téléphoniques par le réseau IP. Aujourd'hui, cette technologie est de plus en plus déployée au sein des universités et laboratoires de recherche connectés au Réseau National de télécommunications pour la Technologie l'Enseignement et la Recherche (RENATER) qui est le réseau académique français.

Afin d'interconnecter les systèmes de téléphonie mis en place par les établissements connectés, à RENATER, une maquette expérimentale d'interconnexion des sites a été déployée. Cette maquette repose sur un serveur de routage d'appel inter-site qui utilise le protocole SIP *(Session Initiation Protocol)*. C'est dans ce contexte que j'ai réalisé mon stage de fin d'études.

Après l'étude du fonctionnement de cette maquette et un inventaire de l'état de l'art dans le domaine de la ToIP, la deuxième phase consiste à la mise en place d'un service pilote de routage d'appels pour la communauté RENATER.

Les objectifs de mon projet de fin étude étaient :

- Etude des évolutions possibles de la maquette pour la mise en place du service pilote.
- Etude et mise en place d'une solution de supervision du service pilote.
- Etude et mise en place d'une solution de comptabilisation d'appels.
- Sécurisation du service pilote.

Ce mémoire est donc l'aboutissement de 4 mois de travail au sein de l'équipe SIPA (Services IP Avancés et prospective) du GIP RENATER.

Mots clés : ToIP, SIP, supervision, comptabilisation d'appels, sécurité

Abstract

The telephony over IP (ToIP) is becoming a new trend in technology widely used nowadays in almost all business sectors. Its concepts rely on transiting the telephone communications through the IP network. Today, this technology is implemented inside a number of universities and research laboratories connected to the National telecommunications Network for Technology, Education and Research (RENATER) which is the French academic network.

In order to interconnect the telephone systems already installed in these academic sites, an experimental testbed has been successfully implemented. This testbed is based on a call routing server using SIP protocol.

After checking this testbed functions and preparing a state of the art on its technologies, a second step consisted to implement a phone call routing pilot service for the (RENATER) community. Based on this context, I achieved my internship graduation.

The main objectives of my internship were:

- to study possible evolutions of the current testbed to deploy the pilot service
- to study and implement a solution for monitoring the current pilot service
- to study and implement a calling accounting system
- Securing the pilot service

This report is the result of 4 months of the work I achieved inside the SIPA team (IP advanced services and prospective) of GIP RENATER.

Keywords: ToIP, SIP, supervision, accounting, security.

Table des matières

Remerciements 2
ملخص 3
Résumé 4
Abstract 5
Table des matières 6
Liste des Figures 8
Liste des Tableaux 9
Glossaire des Acronymes 10
CHAPITRE I : Contexte de travail 14
Introduction 14
1. GIP RENATER 14
2. Réseau RENATER 15
3. Equipe SIPA 17
4. Présentation du stage 17
4.1 Cadre et Objectifs du stage 17
4.2 Planification du projet de stage 18
Conclusion 18
CHAPITRE II : Etat de l'art des protocoles associés à la ToIP 20
Introduction 20
1. Protocoles liés à la ToIP 20
2.1 Signalisation 21
2.1.1 SIP (Session Initiation Protocol) 22
2.2 Transport 26
2. Standard ENUM 27
3. Problématique ToIP avec les NAT et les pare-feux 28
3.1 Problème de NAT 29
3.2 Problème de pare-feux 29
3.3 Solutions de traversées des NAT et des pare-feux 31
3.3.1 Passerelle de la couche application (ALG) 32
3.3.2 STUN 32
3.3.3 TURN 33
3.3.4 ICE 33
3.3.5 Résumé des solutions 34
4. ToIP et la sécurité des communications voix 34
4.1 Vulnérabilités de la ToIP 35
4.2 Exemples d'attaques sur l'infrastructure ToIP 35
4.3 Solutions de sécurité de la ToIP 36
5. La ToIP et IPv6 36
Conclusion 37

CHAPITRE III : Design et ingénierie ToIP 39
Introduction 39
A. Présentation de l'existant 39
1. Description de la maquette ToIP de RENATER 39
1.1 Architecture de la maquette 39
1.2 Principe de fonctionnement 40
2. Présentation d'OpenSER 42
B. Travail effectué 43
1. Évolutions de la plate-forme de routage d'appels OpenSER 43
1.1 Objectifs 43
1.2 Etude 43
1.3 Evolutions 44
2. Mise en place d'une solution de supervision 44
2.1 Objectifs 44
2.2 Etude 44
2.3 Solution Nagios 45
3. Mise en place d'une solution de Comptabilisation d'appels 48
3.1 Objectifs 48
3.2 Etude 48
3.3 Mise en place 48
3.4 Développement d'une interface web 51
3. Sécurisation du pilote ToIP 53
4. Gestion de l'accessibilité des sites 54
5. Tests de validation 55
Conclusion 56
Conclusion générale 57
Références bibliographiques 58
ANNEXES 59

Liste des Figures

Figure 1 : organigramme RENATER 15
Figure 2 : architecture du réseau RENATER 16
Figure 3 : services RENATER 16
Figure 4 : domaines de compétence de l'équipe SIPA 17
Figure 5 : planning de déroulement du stage 18
Figure 6 : pile SIP 22
Figure 7 : architecture SIP 23
Figure 8 : exemple d'une communication SIP 26
Figure 9 : principe de fonctionnement d'ENUM 27
Figure 10 : exemple d'utilisation d'ENUM 28
Figure 11 : problème de NAT avec SIP 29
Figure 12 : problème du firewall avec la ToIP 30
Figure 13 : message INVITE 31
Figure 14 : message 200 OK 31
Figure 15 : technique de la passerelle d'application 32
Figure 16 : principe de fonctionnement du protocole STUN 33
Figure 17 : principe de fonctionnement du protocole TURN 33
Figure 18 : maquette expérimentale ToIP de RENATER 40
Figure 19 : principe de fonctionnement de la maquette 41
Figure 20 : principe de fonctionnement du serveur OpenSER 41
Figure 21 : composantes d'OpenSER 42
Figure 22 : architecture de base de Nagios 45
Figure 23 : interface de l'état des services supervisés 46
Figure 24 : interface du temps de réponse du service Ping 47
Figure 25 : interface de l'état du service SIP de serveur Kamailio dans le temps 47
Figure 26 : architecture de la solution mise en place pour la supervision 47
Figure 27 : principe de fonctionnement de FreeRADIUS avec Kamailio 49
Figure 28 : principe de fonctionnement de module ACC avec MySQL 50
Figure 29 : organigramme de la solution de comptabilisation d'appels 50
Figure 30 : architecture de la solution de comptabilisation des appels 51
Figure 31 : la page d'accueil de l'application 52
Figure 32 : ecran des détails des appels 52
Figure 33 : ecran de nombre des appels par jour 52
Figure 34 : ecran de rechercher sur les appels 53
Figure 35 : organigramme de la solution de sécurisation du service pilote 54

Liste des Tableaux

Tableau 1 : comparatif du norme SIP et H323 21
Tableau 2 : résumé des solutions de traversées les NAT et les pare-feux..... 34
Tableau 3 : résumé des résultats de comparaison entre Kamailio et OpenSIPS 43
Tableau 4 : les résultats des tests de la gestion des messages d'erreur 55
Tableau 5 : les résultats des tests de validation 55

Glossaire des Acronymes

A

ALG	Application Layer Gateway

C

CDR	Call Detail Record
CGI	Common Gateway Interface
CRIHAN	Centre de Ressources Informatiques de Haute- Normandie

D

DNS	Domain Name System
DoS	Denial of Service

E

ENUM	tElephone NUmber Mapping

G

GNU	GNU is not Unix
GPL	General Public Licence

H

HTTP	HyperText Transfer Protocol

I

ICE	Interactive Connectivity Establishment
IETF	Internet Engineering Task Force
INRIA	Institut National de Recherche en Informatique et en Automatique
IP	Internet Protocol
IPBX	Internet Protocol Private Branch eXchange
IPsec	Internet Protocol Security
IPv6	Internet Protocol version 6

L

L2VPN	Layer 2 Virtual Private Networks

M

MGCP	Media Gateway Control Protocol

N

NAT	Network Address Translation

P

PHP	PHP: Hypertext Preprocessor
PNP	PNP is NOT Perfparse

R

RADIUS	Remote Authentication Dial-In User Service
RFC	Request For Comment
RTC	Réseau téléphonique commuté
RTCP	Real-Time Control Protocol
RTP	Real-Time transport Protocol

S

SCCP	Skinny Client Control Protocol
SDP	Session Description Protocol
SER	SIP Express Router
SIP	Session Initiation Protocol
SIPS	Session Initiation Protocol Secure
SRTP	Secure Real-time Transport Protocol
SSH	Secure Shell
STUN	Simple Traversal of UDP Trough NAT

T

ToIP	Telephony over Internet Protocol
TURN	Traversal Using Relay NAT

U

UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UIT-T	Union Internationale des Télécommunications - normalisation des Télécommunications

V

VLAN	Virtual Local Area Network
VoIP	Voice over Internet Protocol

Introduction générale

La téléphonie sur IP constitue actuellement une des plus importantes évolutions dans le domaine des Télécommunications. Il y a quelques années, la transmission de la voix sur le réseau téléphonique classique ou RTC constituait l'exclusivité des télécommunications. Aujourd'hui, la donne a changé. La transmission de la voix via les réseaux IP constitue une nouvelle évolution majeure comparable à la précédente. Au delà de la nouveauté technique, la possibilité de fusion des réseaux IP et téléphoniques entraîne non seulement une diminution de la logistique nécessaire à la gestion des deux réseaux, mais aussi une baisse importante des coûts de communication ainsi que la possibilité de mise en place de nouveaux services utilisant simultanément la voix et les données.

Le travail présenté dans ce rapport entre dans le cadre de mon projet de fin d'études en cycle d'ingénieur, option Génie Réseaux et Systèmes, à l'Ecole Nationale des Sciences Appliquées de Marrakech. Je l'ai effectué au groupement d'Intérêt public RENATER (GIP RENATER) à Paris. Le sujet était la mise en œuvre du service pilote ToIP de RENATER.

Au long de ce rapport, je vais résumer mon stage en 4 chapitres principaux.

Le 1er chapitre présentera l'organisme d'accueil et le sujet du stage de fin d'études et ses objectifs.

Le 2ème chapitre donnera un aperçu global sur les protocoles associés à la téléphonie sur IP.

Le 3ème chapitre sera consacré à la présentation de la maquette expérimentale ToIP existante.

Le dernier chapitre présentera le travail effectué pendant le stage, à savoir la mise en place du service pilote et des solutions qui permettent de comptabiliser les appels, de superviser et sécuriser ce service.

Je finirai ce rapport par une conclusion générale et des perspectives.

Chapitre 1

CHAPITRE I : Contexte de travail

Introduction

Ce chapitre présente d'une manière générale le contexte de travail et les objectifs de mon projet de fin d'études.

Je vais commencer par une présentation du groupement d'intérêt public RENATER comme étant l'organisme d'accueil, après je vais présenter le réseau RENATER (Réseau National de Télécommunications pour la Technologie, l'Enseignement et la Recherche). Ensuite, je vais présenter SIPA (Services IP Avancés et prospective) qui est l'équipe que j'ai intégrée pendant mon stage. Enfin je vais donner une description de mon projet de fin d'études, et ses objectifs.

1. GIP RENATER

Crée en 1993, Le GIP RENATER réunit de grands organismes de recherche et d'enseignement, ainsi que le ministère en charge de l'enseignement supérieur et de la recherche, pour développer et faire fonctionner le réseau RENATER [1].

GIP (groupement d'Intérêt public) est un organisme à but non lucratif, réunissant des administrations de l'Etat et des organismes publics pour une activité définie : dans le cas du GIP RENATER il s'agit du réseau RENATER.

Le GIP RENATER est le maître d'ouvrage de la partie commune de RENATER, constituée de son épine dorsale RENATER, des liaisons internationales, de ses actions pilotes, et du service SFINX, qui est un GIX (Global Internet eXchange), point d'échange de trafic Internet entre prestataires de services Internet, ou opérateurs de télécommunications qui veulent échanger du trafic IP, sans transit et sans passer par des infrastructures internationales.

Le GIP RENATER est également le coordinateur technique et opérationnel global de l'ensemble du réseau RENATER y compris ses éléments régionaux. Il représente le réseau RENATER auprès des institutions françaises et étrangères, et notamment auprès des autres réseaux de la Recherche.

Le directeur du GIP RENATER est M. Dany Vandromme, professeur des universités ; L'équipe du GIP RENATER comprend aujourd'hui environ 30 personnes : ingénieurs, techniciens et personnels administratifs répartis entre Paris, Montpellier et Rennes.

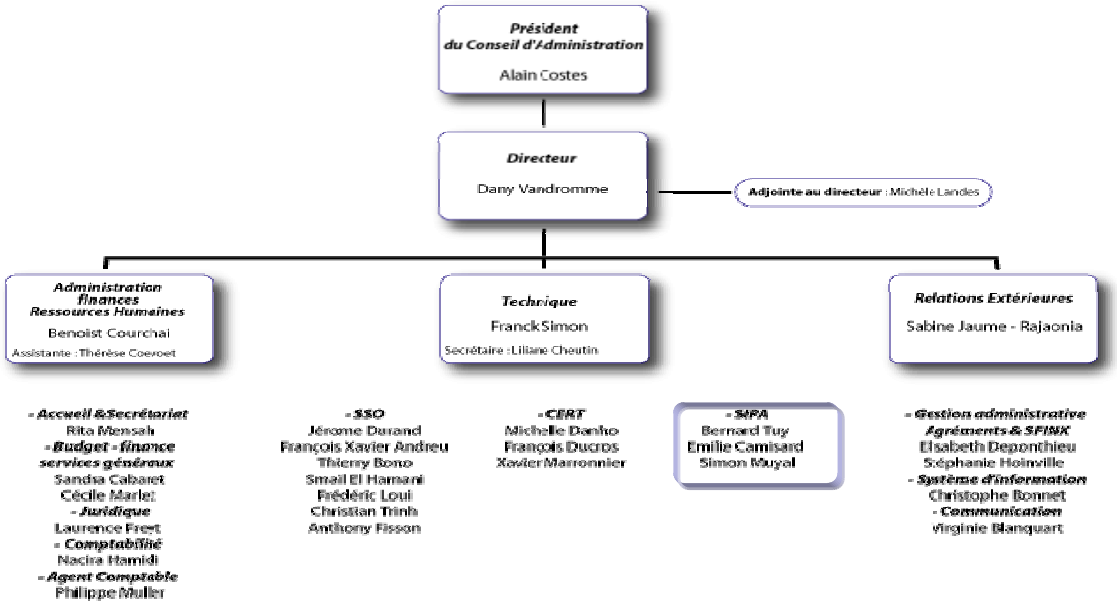

Figure 1 : organigramme RENATER

2. Réseau RENATER

RENATER a été créé dans les années 1990 dans le but de fédérer et d'organiser les infrastructures de télécommunications pour l'Education, la Recherche et l'Enseignement [1].

Aujourd'hui plus de 1000 établissements ayant une activité dans les domaines de la Recherche, la Technologie et l'Enseignement sont raccordés à RENATER. Ce réseau leur permet de communiquer entre eux, d'accéder aux centres de recherche et aux établissements d'enseignement du monde entier et à l'Internet [1].

Le réseau RENATER est constitué d'une infrastructure nationale reliant des points de présence en région et dans les DOM-TOM ainsi que des liaisons internationales, et un nœud d'échange entre prestataires de service Internet appelé SFINX (Service for French Internet eXchange).

La figure ci-dessous décrit l'architecture globale du réseau RENATER :

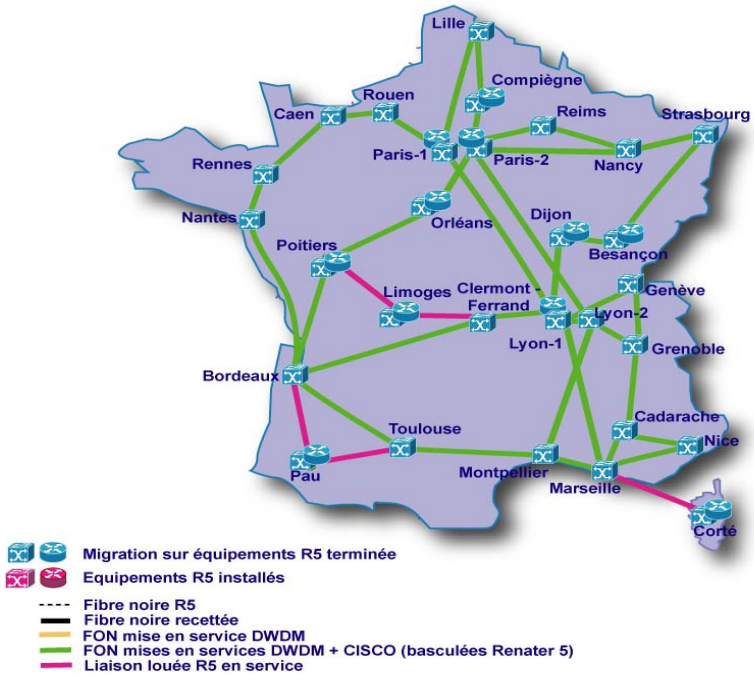

Figure 2 : architecture du réseau RENATER

RENATER propose à ses communautés un large éventail de services [1]:

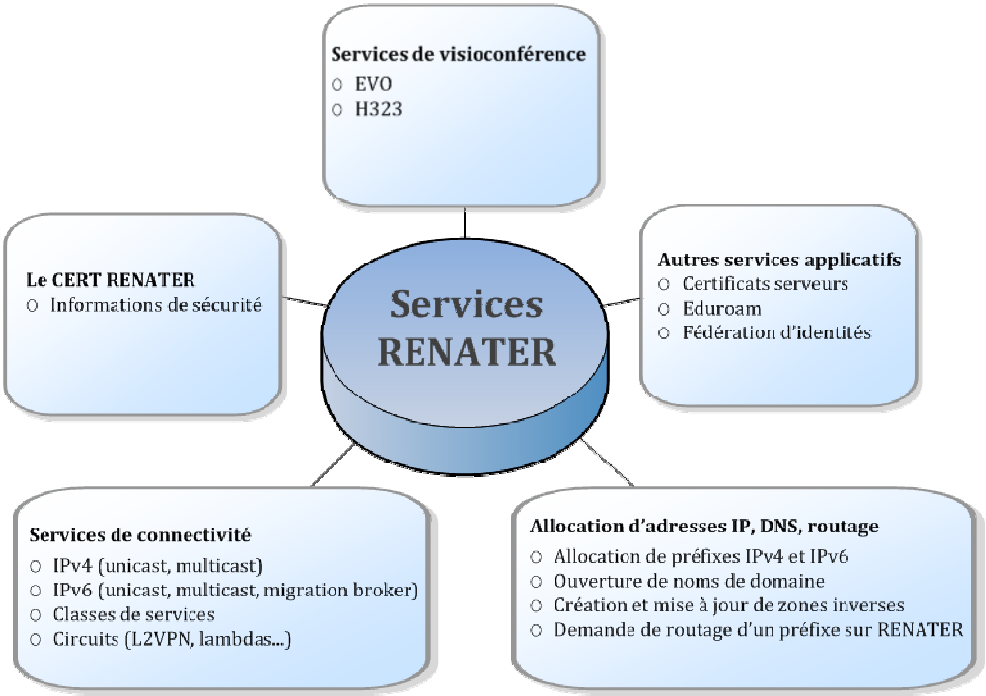

Figure 3 : services RENATER

3. Equipe SIPA

GIP RENATER comprend un certain nombre d'équipe, SIPA en fait partie. L'équipe SIPA (Services IP Avancés et prospective) a une mission de veille technologique. Sa démarche est avant tout expérimentale pour valider les nouveaux protocoles et leurs usages dans des applications ou services émergents.

Le schéma ci-dessous, présente les différents domaines dans lesquels l'équipe SIPA est investie.

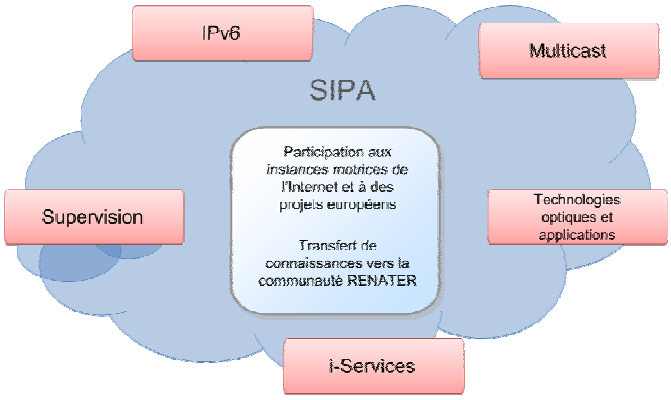

Figure 4 : domaines de compétence de l'équipe SIPA

4. Présentation du stage

4.1 Cadre et Objectifs du stage

La téléphonie sur IP (ToIP) est un service qui est de plus en plus déployé au sein des universités et laboratoires de recherche connectés au réseau RENATER. Une maquette expérimentale reliant quelques sites a été mise en place. Cette maquette repose sur le protocole SIP et le routeur d'appels SIP OpenSER. La maquette permet l'interconnexion des IPBX des sites et le routage d'appels inter-site. Au sein de l'équipe SIPA, l'objectif principal du stage était la mise en place d'un service pilote de ToIP en se basant sur la maquette expérimentale existante.

La première partie du stage consiste à étudier les évolutions possibles de cette maquette pour la mise en place d'un service pilote de routage d'appels au profit des usagers de RENATER, la deuxième partie du stage concerne l'étude et la mise en place d'une solution de supervision du service de routage d'appels, la troisième partie du stage concerne l'étude et la mise en place d'une solution de comptabilisation des appels, et la quatrième partie du stage consiste à la sécurisation du routeur d'appels.

4.2 Planification du projet de stage

La planification est parmi les phases d'avant projet les plus importantes. Elle consiste à déterminer et à ordonnancer les tâches du projet et à estimer leurs charges respectives. Parmi les outils de planification de projet, j'ai utilisé le diagramme de GANTT, c'est un outil qui permet de planifier le projet et de rendre plus simple le suivi de son avancement. Ce diagramme permet aussi de visualiser l'enchainement et la durée des différentes tâches durant le stage comme il est illustré par la figure qui suit :

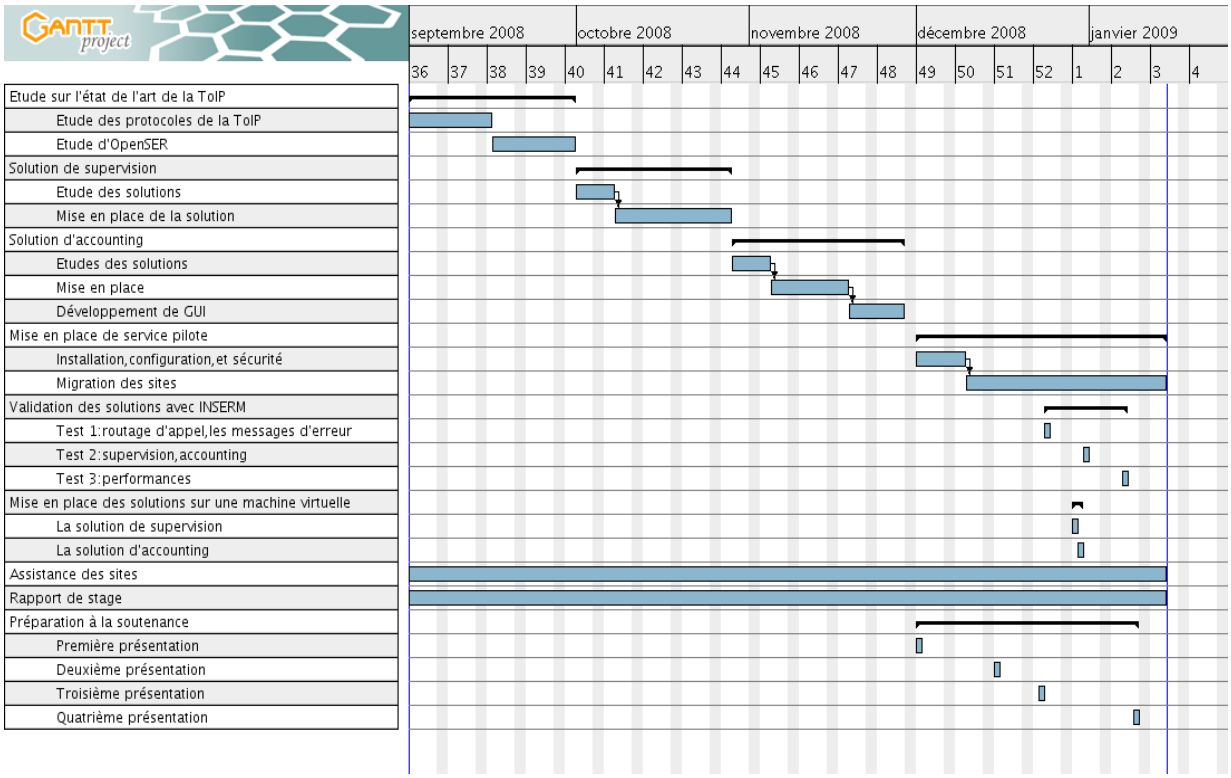

Figure 5 : planning de déroulement du stage

Conclusion

Ce chapitre introductif a été consacré essentiellement à la présentation de l'environnement dans lequel mon projet de fin d'études a été effectué. Elle a aussi mis l'accent sur la présentation du contexte de mon projet, ses objectifs et sa planification.

Chapitre 2

CHAPITRE II : Etat de l'art des protocoles ToIP

Introduction

La téléphonie sur IP est une technologie de communication vocale en pleine émergence. Elle fait partie d'un tournant dans le monde de la communication. En effet, la convergence du triple play (voix, données et vidéo) fait partie des enjeux principaux des acteurs de la télécommunication aujourd'hui.

La ToIP possède actuellement une véritable opportunité économiques pour les entreprises telles que la diminution du coût en infrastructure, de la facture de téléphone. Elle permet l'intégration de nombreux services. La téléphonie sur IP est basée sur des standards ouverts : elle permet donc l'interaction avec les équipements téléphoniques standards. Toutefois, les aspects techniques sous-jacents à cette nouvelle technologie ne sont pas toujours bien maîtrisés. Les problèmes dus au NAT, les pare-feux, la sécurité, etc. sont des problèmes qui restent encore à dominer.

Ce chapitre est consacré à l'étude des protocoles associés à la téléphonie sur IP. Cette étude va me permettre par la suite de mener à bien le projet. Pour cela, je commence tout d'abord par présenter les principaux protocoles de signalisation et de transport de la ToIP et le protocole ENUM dont l'usage est encore incertain au sein des opérateurs de ToIP. Je traite ensuite les problèmes posés par les NAT et les pare-feux dans une architecture de ToIP et les exemples de solutions pour résoudre ces problèmes. Je présente ensuite les vulnérabilités spécifiques à la ToIP et les mécanismes de sécurité. Je termine en présentant l'état de l'art du déploiement du protocole IPv6 dans la ToIP.

1. Protocoles liés à la ToIP

La téléphonie sur IP ou ToIP (Telephony over IP) est un service de téléphonie qui transporte les flux voix des communications téléphoniques sur un réseau IP. A la différence de la VoIP où l'on ne fait qu'établir une communication « voix », la ToIP intègre l'ensemble des services associés à la téléphonie : double appel, messagerie, renvoie d'appel, FAX, etc.

Afin de rendre possibles les communications ToIP, les solutions proposées dopent la couche IP par des mécanismes supplémentaires nécessaires pour apporter la QOS nécessaire au flux voix de types temps réel, en plus de l'intelligence nécessaire à l'exécution de services. A cet effet, il existe deux types de protocoles principaux utilisés dans la ToIP :

- Protocoles de signalisation
- Protocoles de transport

2.1 Signalisation

La signalisation correspond à la gestion des sessions de communication (ouverture, fermeture, etc.). Le protocole de signalisation permet de véhiculer un certain nombre d'informations notamment:

- Le type de demande (enregistrement d'un utilisateur, invitation à une session multimédia, annulation d'un appel, réponse à une requête, etc.).
- Le destinataire d'un appel.
- L'émetteur.
- Le chemin suivi par le message.

Plusieurs normes et protocoles ont été développés pour la signalisation ToIP, quelques uns sont propriétaires et d'autres sont des standards. Ainsi, les principales propositions disponibles pour l'établissement de connexions en ToIP sont :

- SIP (Session Initiation Protocol) qui est un standard IETF (Internet Engineering Task Force) décrit dans le RFC 3261.

- H323 englobe un ensemble de protocoles de communication développés par l'UIT-T (Union Internationale des Télécommunications – secteur de la normalisation des Télécommunications).

- MGCP (Media Gateway Control Protocol) standardisé par l'IETF (RFC 3435).

- SCCP (Skinny Client Control Protocol) est un protocole propriétaire CISCO.

Aujourd'hui le plus répandu d'entre eux est le SIP, ce protocole est largement déployé et utilisé au sein de RENATER. Le tableau qui suit dresse un léger comparatif entre la norme SIP et H323.

	SIP	H323
Nombre d'échanges pour établir la connexion	3 à 5 Aller-retour	6 à 7 Aller-retour
Maintenance du protocole	Simple (texte comme HTTP)	Complexe
Evolution	Ouvert à de nouvelles fonctions	Ajout d'extensions propriétaires
Multicast	Oui	Oui

Tableau 1 : comparatif du norme SIP et H323

2.1.1 SIP (Session Initiation Protocol)

- **Introduction**

Le protocole SIP (Session Initialisation Protocol) a été initié par le groupe MMUSIC (Multiparty Multimedia Session Control) [RFC 2543] et désormais repris et maintenu par le Groupe SIP de l'IETF [RFC 3261]. SIP est un protocole de signalisation appartenant à la couche application du modèle OSI. Il a été conçu pour l'ouverture, le maintient et la terminaison de sessions de communications interactives entre des utilisateurs. De telles sessions permettent de réaliser de l'audio, de l'enseignement à distance et de la voix (téléphonie) sur IP essentiellement. Pour l'ouverture d'une session, un utilisateur émet une invitation transportant un descripteur de session permettant aux utilisateurs souhaitant communiquer de négocier sur les algorithmes et codecs à utiliser. SIP permet aussi de relier des stations mobiles en transmettant ou redirigeant les requêtes vers la position courante de la station appelée. Enfin, SIP est indépendant du médium utilisé et aussi du protocole de transport des couches basses.

- **Architecture protocolaire**

SIP est un protocole indépendant des couches de transport, il appartient aux couches applications du modèle OSI. Le SIP gère la signalisation et l'établissement des sessions interactives de communication multimédias et multipartites. Il est aussi basé sur le concept Client / Serveur pour le contrôle d'appels et des services multimédias. Conçu selon un modèle de type IP, il est hautement extensible et assez simple en conception architecturale, de sorte qu'il peut servir de base à la création d'applications et de services. Il est basé sur le protocole HTTP et peut utiliser UDP ou TCP [8].

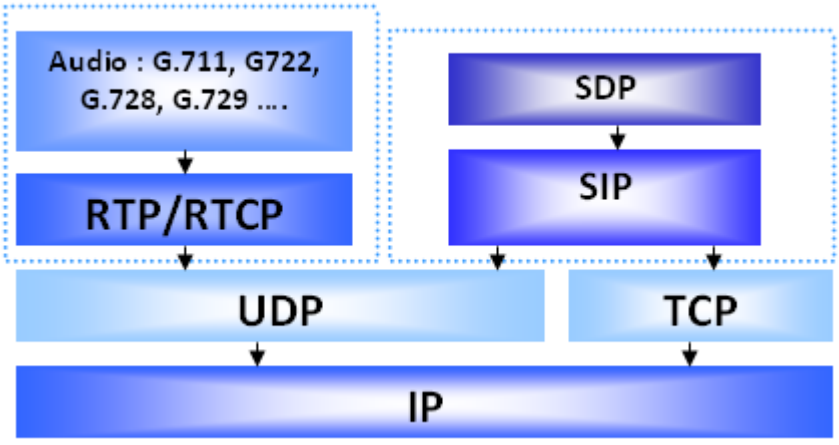

Figure 6 : pile SIP

- **Architecture d'une plate forme SIP**

SIP est un protocole simple et flexible orienté messages. Les principaux composants d'un système basé sur SIP sont :

- **Terminal SIP (User Agent Client ou UAC) :** Peut être aussi bien un SoftPhone (logiciel) qu'un HardPhone (téléphone IP). Les UAC sont capables d'émettre et de recevoir de la signalisation SIP.

- **Proxy Server :** encore appelé serveur mandataire auquel est relié un terminal fixe ou mobile, agit comme serveur envers le client et comme client envers les autres UAS.

- **Redirect Server :** Ce serveur permet de rediriger les appels vers la position courante d'un utilisateur. Il réalise simplement une association d'adresses vers une ou plusieurs nouvelles adresses.

- **Location Server :** Il fournit la position courante des utilisateurs dont la communication traverse les serveurs mandataire et de redirection auxquels il est rattaché.

- **Registrar Server :** Ce serveur reçoit et accepte les inscriptions des utilisateurs (adresse IP, port, login).

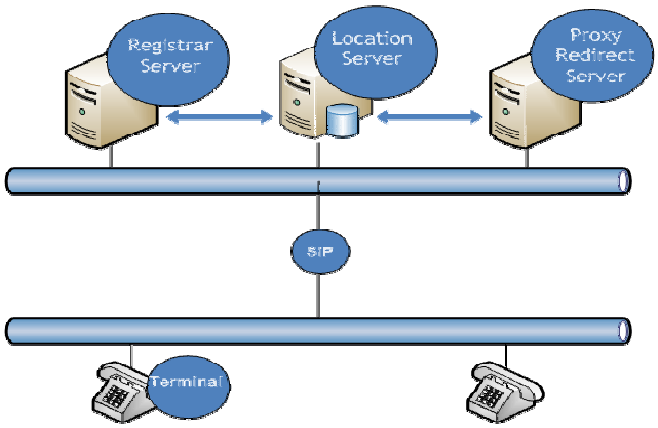

Figure 7 : architecture SIP

- **Structure des messages SIP**

Les messages SIP sont caractérisés par une ligne de début, plusieurs entêtes et le corps du message [8].

➢ Les entêtes des messages SIP

Les entêtes ont pour rôle de fournir des informations sur le message et de permettre le traitement du message. A cet effet, le protocole SIP est doté d'un certain nombre d'en-tête dont la structure dépend de la nature et du rôle de chaque en-tête. La structure générale d'un entête est articulée autour de plusieurs champs et chaque champ obéit à un format général : nom_du_champ : valeur_du_champ. Les types d'entête utilisés par les messages du protocole SIP sont au nombre de quatre:

☑ L'entête général

Il est toujours présent et contient les informations de base permettant le traitement du message.

Il a des champs obligatoires suivants :

- Via : il identifie l'entité de relais. En effet, chaque entité qui émet ou relaye un message SIP insère son identité afin de prévenir les boucles et indiquer le chemin de réponse.
- From : il identifie l'initiateur de la requête.
- To : il identifie le destinataire de la requête.
- Call-Id : c'est l'identificateur unique de la session.
- Cseq : il identifie la séquence d'un appel : par exemple plusieurs messages « invite » avec de Cseq différents.

☑ L'entête de requête

Cet en-tête est non toujours utilisé. Il contient des informations supplémentaires à destination du serveur SIP permettant le traitement de la requête par celui-ci.

☑ L'entête de réponse

Cet en-tête est non toujours utilisé tout comme l'entête de requête. Il contient des informations supplémentaires ajoutées par le serveur SIP permettant le traitement de la réponse.

☑ L'entête d'entité

Cet en-tête est toujours utilisé. Son rôle est de définir le type et le format des informations contenues dans les messages.

➢ Le corps du message

Il fournit suffisamment d'informations pour permettre la participation à une session multimédia. Ces informations sont : le codec, destination (adresse IP et port UDP), nom de la session, etc. Le message du corps est codé conformément au protocole SDP (Session Description Protocol). SDP est sans doute le protocole le plus important de l'architecture SIP, SDP a fait l'objet de la proposition de norme RFC 2327. C'est un protocole dont l'objectif est d'établir un descripteur de sessions multimédia à ouvrir, il porte les informations suivantes :

☑ Adresses de destination *SIP: //UserX@ensa.ac.ma*.
☑ Algorithmes de codage Audio et Vidéo.
☑ Type de trafic RTP.

- **Requêtes et Réponses SIP**

SIP est un protocole de type client serveur. A cet effet, les échanges entre un terminal appelant et un terminal appelé se font par l'intermédiaire de requêtes et réponse SIP.

Voici une liste exhaustive des requêtes SIP :

- INVITE : Cette requête indique que l'application (ou utilisateur) correspondante à l'Url SIP spécifié est invitée à participer à une session.
- ACK : Cette requête permet de confirmer que le terminal appelant a bien reçu une réponse définitive à une requête INVITE.
- BYE : Cette requête est utilisée par le terminal de l'appelé pour signaler qu'il souhaite mettre un terme à la session.
- CANCEL – Cette requête est envoyée par un terminal ou un serveur mandataire afin d'annuler une requête non validée par une réponse finale.
- REGISTER – Cette méthode est utilisée par le client pour enregistrer l'adresse listée dans le champ TO par le serveur auquel il est relié.
- OPTIONS – Un serveur mandataire en mesure de contacter le terminal appelé, doit répondre à une requête OPTIONS en précisant ses capacités à contacter le même terminal.

A ces requêtes sont associées des réponses qui sont dans le même format que celles du protocole HTTP. Voici les plus importantes d'entre elles :

- 1XX – messages d'informations (100 – essai, 180 – sonnerie, 183 – en cours)
- 2XX – succès de la requête (200 –OK)
- 3XX – Redirection de l'appel, la demande doit être dirigée ailleurs
- 4XX – Erreur du client (La requête contient une syntaxe erronée)
- 5XX – Erreur du serveur (le serveur n'a pas réussi à traiter une requête correcte)
- 6XX – Echec général (606 – requête non acceptable par aucun serveur)

- **Fonctionnement**

SIP intervient aux différentes phases de l'appel :

- Localisation du terminal de l'interlocuteur.

- Analyse du profil et des ressources du destinataire.
- Négociation du type de média (voix, audio, vidéo…) et des paramètres de communication.
- Disponibilité du correspondant, détermine si le poste appelé souhaite communiquer, et autorise l'appelant à le contacter.
- Etablissement et suivi de l'appel, avertit les parties appelant et appelé de la demande d'ouverture de session, gestion du transfert et de la fermeture des appels.
- Gestion de fonctions évoluées : retour d'erreurs, …

Le schéma suivant illustre le scénario d'une communication SIP.

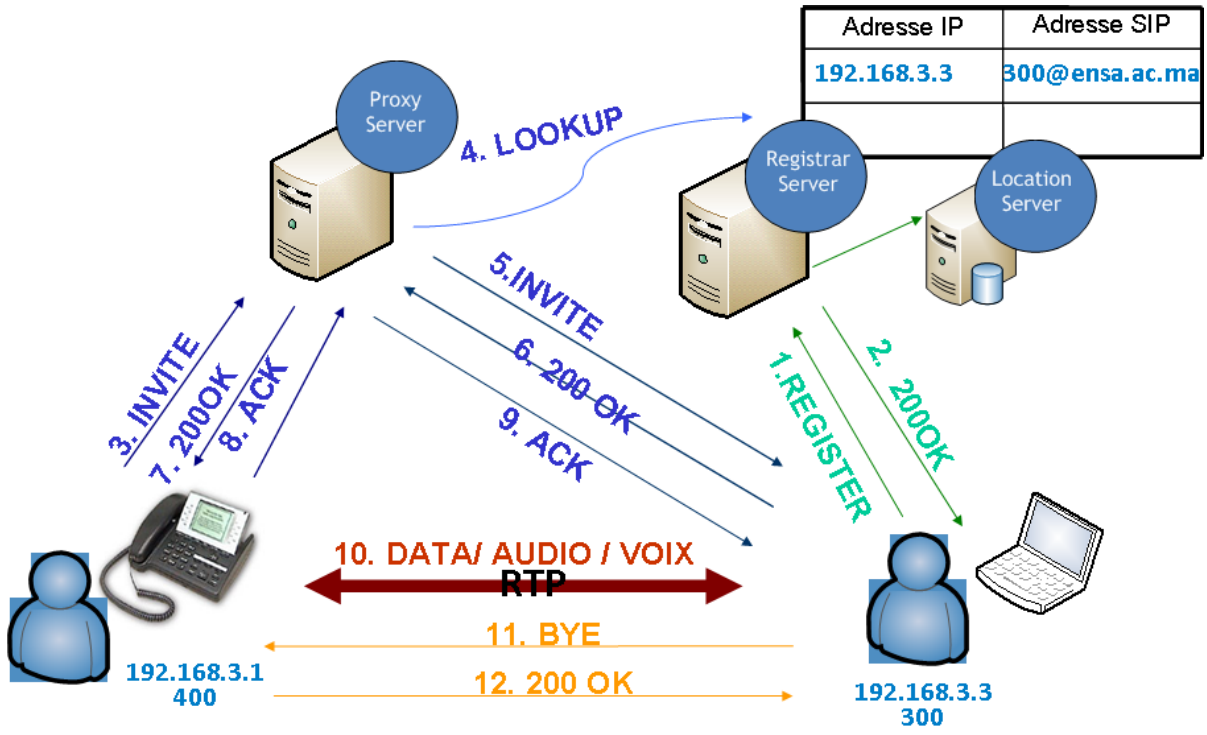

Figure 8 : exemple d'une communication SIP

2.2 Transport

Lors d'une communication ToIP, une fois la phase de signalisation réalisée, la phase de communication est initiée. Dans cette phase, un protocole de transport permet d'acheminer les données voix entre plusieurs utilisateurs vu que la couche TCP propose un transport fiable mais lent, et la couche UDP un transport rapide mais non fiable. La communauté IETF a mis en place un nouveau couple de protocole RTP (Real-Time transport Protocol) et RTCP

(Real-Time Control Protocol) pour apporter la fiabilité à l'UDP tout en exploitant sa rapidité. RTP et RTCP sont les deux protocoles qui sont principalement utilisés pour le transport de flux média sur le réseau IP. RTP permet de transporter les données entre plusieurs utilisateurs en plus de la gestion temps réelle des sessions. Tandis que, RTCP est utilisé pour transmettre régulièrement des paquets de contrôle, qui contiennent diverses statistiques, ce qui permet de vérifier la qualité de transmission.

2. Standard ENUM

La fourniture à grande échelle du service ToIP suppose l'identification aisée des terminaux IP connectés au réseau désireux accéder à ce service. Cette identification est basiquement faite à travers les adresses IP. Afin d'étendre les possibilités d'adressage l'UIT-T a travaillé sur le standard ENUM.

ENUM (tElephone NUmber Mapping) est un protocole défini par l'IETF dans le RFC 3761[11]permettant d'utiliser un numéro de téléphone (E.164[1]) comme clé de recherche dans le DNS pour trouver la manière de joindre une personne (par exemple : n° de téléphone mobile, n° de fax, adresse de téléphonie IP, adresse e-mail, adresse de messagerie instantanée, etc.)

Le principe d'ENUM repose sur la création d'un nom de domaine Internet pour chaque numéro de téléphone du plan de numérotation international E.164. Les coordonnées que les utilisateurs souhaitent publier pour leur propre numéro de téléphone sont ensuite "stockées" dans le système des noms de domaine Internet (DNS) et ainsi rendues accessibles de manière globale pour tous.

Voici un schéma expliquant le principe de fonctionnement d'ENUM :

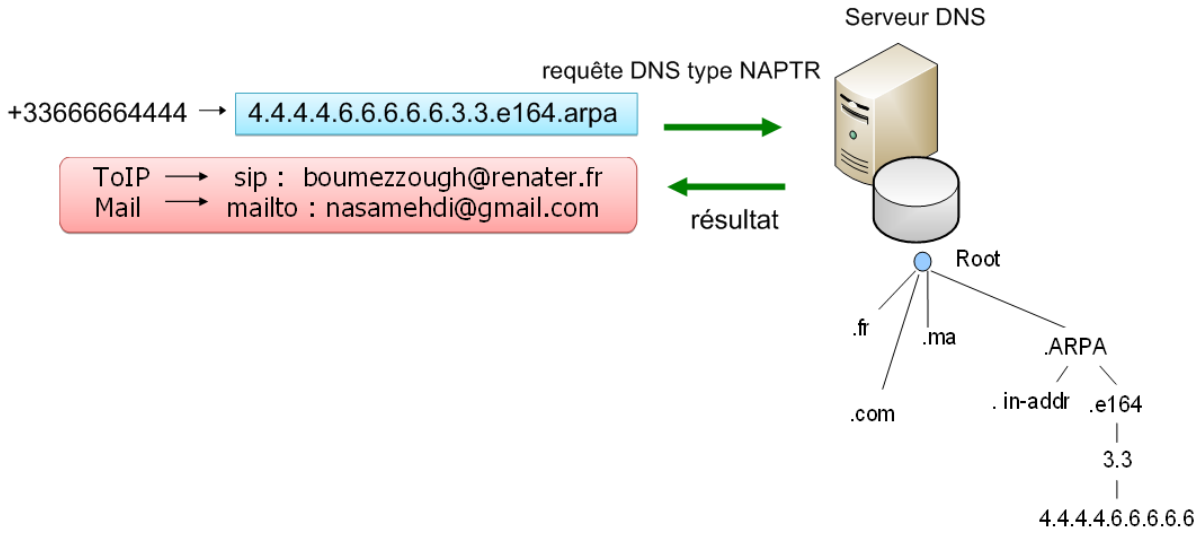

Figure 9 : principe de fonctionnement d'ENUM

[1] E.164 est le nom de la norme de l'UIT qui standardise les numéros de téléphone au niveau mondial.

Dans ce schéma le numéro +33666664444 est transformé en nom de domaine en l'inversant (comme on fait pour trouver un nom de domaine à partir d'une adresse IP), cela donnerait 4.4.4.4.6.6.6.6.6.3.3.e164.arpa. On cherche alors les enregistrements NAPTR[2] pour ce nom de domaine. Dans cet exemple, le titulaire du numéro de téléphone +33666664444 peut être joint en SIP en utilisant l'URI sip:boumezzough@renater.fr et par courrier électronique à nasamehdi@gmail.com.

L'avantage d'ENUM serait de joindre une personne avec un seul numéro sur différents services de communication aussi à travers ENUM une personne peut très bien spécifier l'ordre de préférence des services et des terminaux à utiliser.

La figure ci-dessous présente un exemple d'utilisation d'ENUM.

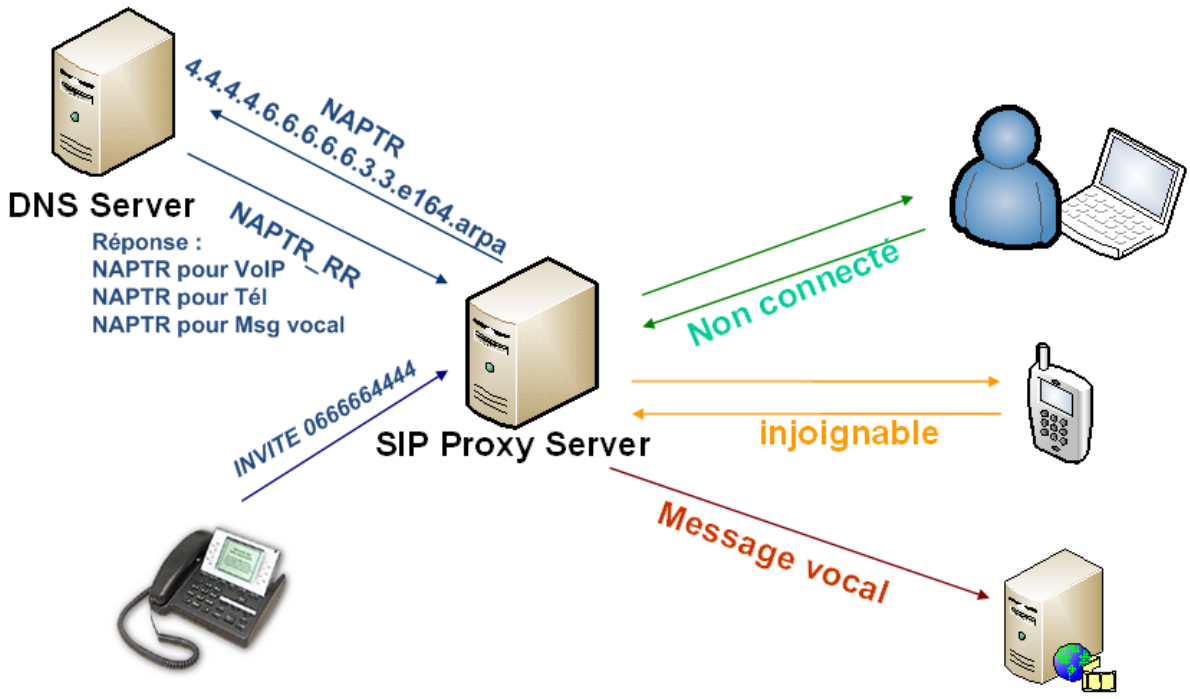

Figure 10 : exemple d'utilisation d'ENUM

3. Problématique ToIP avec les NAT et les pare-feux

Dans une architecture ToIP, les NAT et les pare-feux représentent un problème pour les flux de signalisation et média. L'objectif de ce paragraphe est de comprendre cette problématique et de présenter des exemples de solutions existantes pour résoudre ce problème.

[2] NAPTR record ou Name Authority Pointer record qui donne accès à des règles de réécriture de l'information, permettant des correspondances entre un nom de domaine et une ressource. Il est spécifié dans la RFC 3403.

3.1 Problème de NAT

Le mécanisme NAT (Network Address Translation) est défini dans le RFC 1631[12]. Le NAT permet de faire correspondre les adresses IP internes souvent non routables d'un domaine à un ensemble d'adresses routables. Avec la ToIP, ce mécanisme représente un problème pour le transit des flux multimédia.

En effet, les informations utilisées pour la signalisation ou la communication sont incluses au niveau 4 et les couches supérieures du modèle OSI, tandis que les NAT travaillent sur la couche 3.

Voici un schéma expliquant le problème de NAT avec la ToIP :

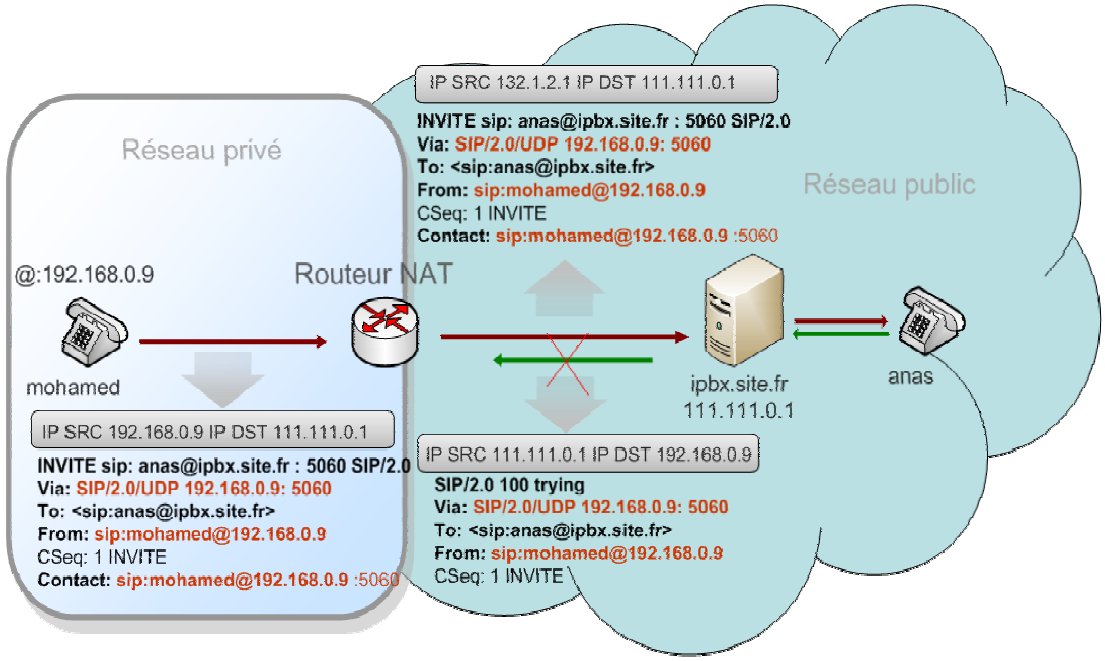

Figure 11 : problème de NAT avec SIP

Dans cet exemple, Mohamed ne pourra établir de communication avec Anas étant donné que l'IPBX n'arrive pas à relayer les réponses SIP de Anas. En effet, lors de la traduction d'adresse effectuée par le routeur NAT, seuls l'adresse et le numéro de port contenus dans l'entête du paquet IP sont modifiés. L'adresse et le numéro de port contenus dans le corps de la requête INVITE du message SIP ne sont pas modifiés. Hors cette adresse est non routable.

3.2 Problème de pare-feux

Un pare-feu (firewall) est un équipement permettant d'assurer la sécurité d'un site en filtrant le trafic non désiré, il permet de filtrer les paquets venant du réseau public.

Le mode de fonctionnement de la plupart des pare-feux pose un problème pour l'établissement des communications SIP.

SIP, par son fonctionnement interne qui permet la localisation des utilisateurs au sein d'un réseau et la négociation des paramètres de la session (codecs, port RTP, etc.), pose des problèmes pour les flux multimédias qui traversent les firewalls. En effet, dans l'architecture de SIP, plusieurs informations critiques telles que l'adresse IP ainsi que le port à utiliser sont contenues dans les messages SIP.

La problématique engendrée par l'utilisation de SIP au travers des pare-feux vient du fait que ceux-ci sont généralement déployés en utilisant des politiques de filtrage qui rejettent tous les paquets qui ne proviennent pas ou qui ne sont pas destinés à une adresse IP et un port définis. Ces politiques, qui sont généralement statiques, ne permettent pas la traversée d'un flux de données à des protocoles comme SIP qui peut négocier des adresses IP et des numéros du port inconnus par le pare-feu lors de l'établissement de session.

Afin de bien comprendre la problématique engendrée par les pare-feux pour les flux multimédias, voici le schéma d'un scénario d'établissement de session et comment le firewall bloque le contenu multimédia.

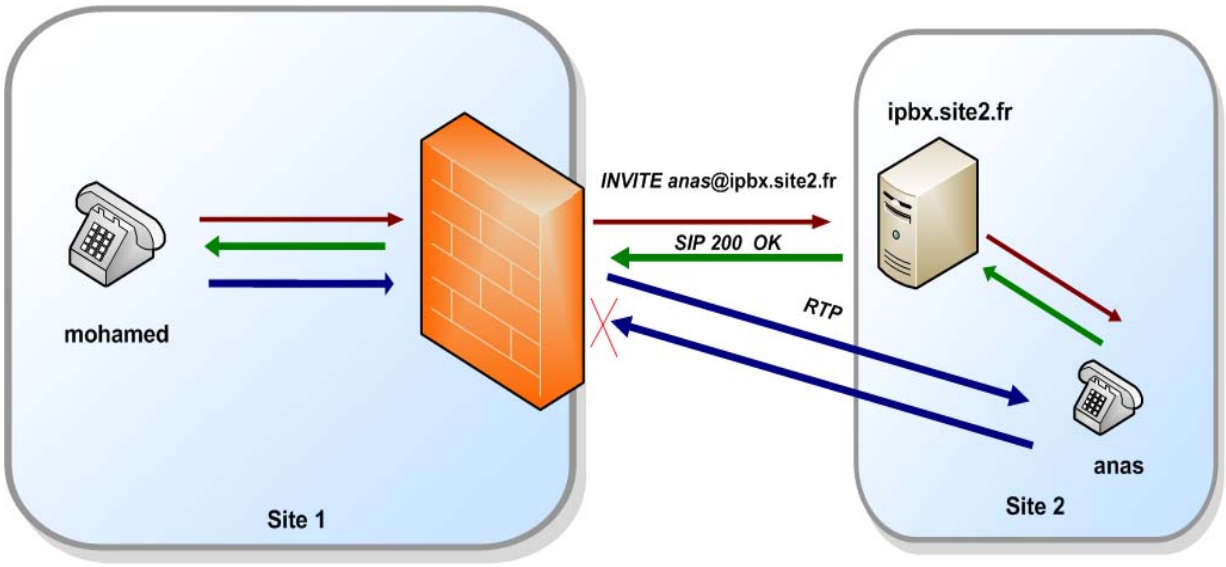

Figure 12 : problème du firewall avec la ToIP

Dans cet exemple, l'utilisateur SIP mohamed@poste.site.fr invite l'utilisateur anas@ipbx.site2.fr afin d'ouvrir une session. Mohamed envoie donc une requête d'invitation INVITE (figure 8) contenant les informations de la session à ouvrir à Anas. Anas répond avec le message 200 OK (figure 9) contenant des informations supplémentaires sur la session à ouvrir.

Comme on peut le constater, l'adresse ainsi que le port à utiliser lors de l'ouverture de la session audio sont contenus dans le corps des messages INVITE et OK. Ces deux informations servent à l'établissement du flux audio entre Mohamed et Anas. Dans notre exemple,

Mohamed et Anas utilisent respectivement les ports 3456 et 5004. Par conséquent, comme le firewall a des règles de filtrage strictes et statiques, le contenu multimédia d'Anas sera donc tout simplement bloqué par le firewall.

Figure 13 : message INVITE

Figure 14 : message 200 OK

3.3 Solutions de traversées des NAT et des pare-feux

Afin de faire face aux problèmes que posent les pare-feux et les NAT, plusieurs solutions ont été proposées notamment ALG, STUN, TURN et ICE.

3.3.1 Passerelle de la couche application (ALG)

La technique de la passerelle applicative consiste à rendre « intelligents » les pare-feux et les routeurs NAT afin qu'ils soient en mesure d'interpréter un protocole spécifique. Plutôt que de vérifier uniquement l'entête du paquet à traiter, les passerelles réalisent une inspection complète des données dans le corps du paquet. Les passerelles agissent donc en tant que relais spécialisés pour un protocole précis (SIP par exemple).

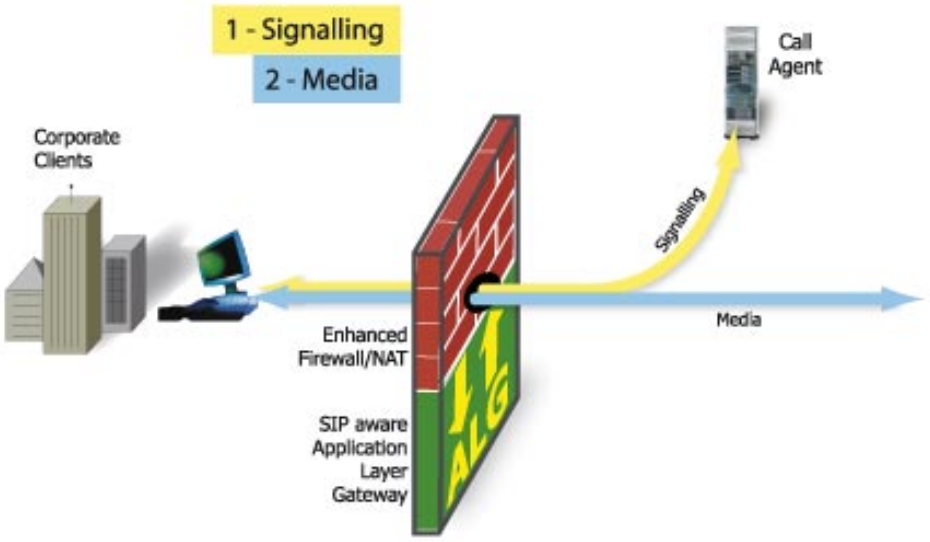

Figure 15 : technique de la passerelle d'application
(*source* : *http://www.newport-networks.com/whitepapers/nat-traversal4.html*)

Le pare-feux/routeur NAT va donc lire le contenu complet du paquet puis va modifier les adresses IP et ports inscrits dans le paquet afin de pouvoir transmettre le paquet dans le réseau public. Suite à cela, le pare-feux/routeur NAT ouvrira un « port d'accès» afin que la communication puisse avoir correctement.

3.3.2 STUN

STUN (Simple Traversal of UDP Through Network Address Translators) est un protocole énoncé dans le RFC 3489 [13] et développé par le groupe de travail de MIDCOM. Cette technique se distingue des techniques des passerelles de la couche application par son indépendance face aux protocoles de communication.

STUN permet de traverser les routeurs NAT en affectant une adresse IP et un numéro de port public à un poste situé dans le réseau privé pour effectuer une communication de type UDP avec un réseau public. Pour ce faire, une série de requêtes à un serveur STUN est effectuée et les réponses du serveur servent à caractériser l'adresse IP et le numéro de port d'un poste à communiquer.

Combinant STUN à des protocoles tel que SIP, plusieurs problèmes reliés aux routeurs NAT peuvent être solutionnés. Seul le cas où des routeurs NAT de type symétrique3 sont utilisés ne peut être traité en utilisant cette technique.

La figure suivante décrit le principe de fonctionnement du protocole STUN.

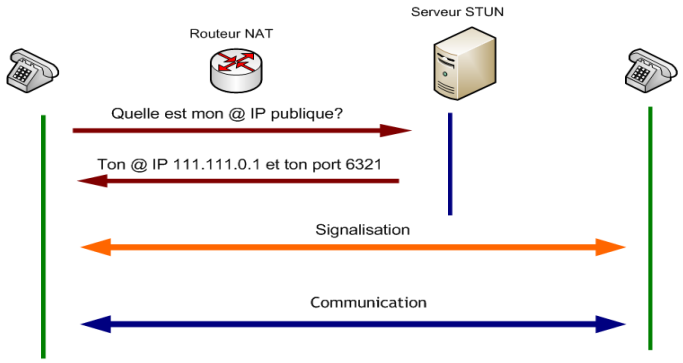

Figure 16 : principe de fonctionnement du protocole STUN

3.3.3 TURN

TURN (Traversal Using Relay NAT) est un mécanisme en cours développement et de standardisation auprès de l'IETF agissant en tant que serveurs de relais. Il a été développé afin de pallier les lacunes du protocole STUN.

Ce protocole permet à des clients qui sont dans des réseaux utilisant des routeurs NAT d'effectuer des connexions entre eux en passant par un serveur de relais.

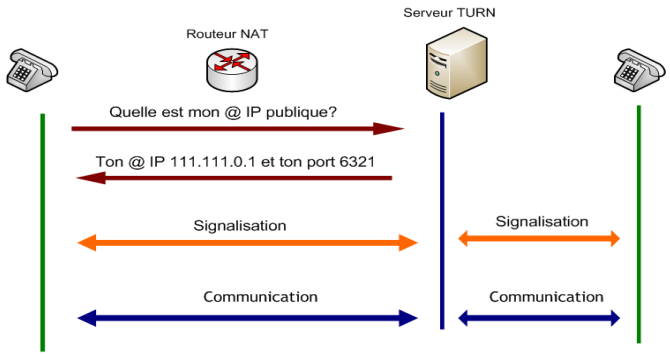

Figure 17 : principe de fonctionnement du protocole TURN

3.3.4 ICE

ICE (Interactive Connectivity Establishment) est une technologie en cours de développement par IETF qui consiste à intégrer STUN et TURN au sein des clients SIP pour déterminer toutes les connexions possibles entre deux postes.

[3] Un NAT symétrique est celui dans lequel la translation d'adresse est calculée en fonction de l'adresse IP et port de la source et de celui du destinataire

3.3.5 Résumé des solutions

Le tableau qui suit résume les principes, avantages et inconvénients des solutions présentées :

solution	principe	Avantages	Inconvénients
ALG	Firewall et routeur NAT « intelligents » capables de travailler au niveau 7	Technique simple	- Temps de latence importants dus au traitement complet et individuel des paquets.
STUN	Détermine le couple (adresses, ports) publics qu'on doit utiliser dans le paquet SIP pour obtenir une réponse	- Protocole standardisé. - Peu d'infrastructure : seul 1 serveur doit être déployé pour effectuer les requêtes.	- il ne fonctionne pas avec les NATs symétriques
TURN	- Basé sur STUN pour l'échange de clés - Le serveur TURN sert de relais entre l'émetteur et le récepteur	- Technique simple	- Modification des programmes nécessaire pour qu'ils puissent intégrer TURN
ICE	- intègre STUN et TURN au sein des clients SIP pour déterminer toutes les connexions possibles entre deux postes.	Traverse tous types de NAT	- Temps d'établissement d'un appel long. - Modification des serveurs et clients pour le déploiement.

Tableau 2 : résumé des solutions de traversées les NAT et les pare-feux

4. ToIP et la sécurité des communications voix

La téléphonie sur IP, malgré ses très nombreux avantages, notamment financiers, comporte des risques majeurs en termes de sécurité des communications voix.

4.1 Vulnérabilités de la ToIP

Un appel téléphonique ToIP se décompose en deux phases : la signalisation qui permet d'établir l'appel, et la phase de transport des flux de medias qui transportent la voix.

Au cours de la phase de signalisation, les messages SIP codés en mode texte sont transmis de façon non chiffrée dans le réseau, ce qui permet à un pirate d'écouter facilement les messages SIP et d'accéder aux informations de transport des flux média.

En outre durant le transport des flux voix, le protocole RTP présente également plusieurs vulnérabilités dues à l'absence d'authentification et de chiffrement. Par voie de conséquence, plusieurs attaques ToIP peuvent avoir lieu.

4.2 Exemples d'attaques sur l'infrastructure ToIP

Il existe de nombreuses attaques possibles sur le réseau ToIP dont les plus répandues, sont :

- Dénis de service (attaque DoS) : l'objectif d'une attaque DoS est de rendre un élément du réseau indisponible. Un exemple de ce type attaque est l'envoi illégitimes de paquets SIP BYE [17].

- Ecoute clandestine : L'objectif de cette attaque est d'écouter le trafic de signalisation et/ou de données, en utilisant des outils d'écoute réseau tels que VOMIT (Voice Over Misconfigured Internet Telephone), SiVuS (SIP Vulnerability Scanner), et WireShark [17].

- Détournement du trafic : l'attaquant redirige à son profit le trafic ToIP. Elle se base sur l'envoi d'un message de redirection indiquant que l'appelé s'est déplacé et donne sa propre adresse comme adresse de renvoie, de cette façon tous les appels destinés a l'utilisateur sont transférés a l'attaquant [17].

- Usurpation d'identité : Ce type d'attaque consiste à usurper l'identité de l'expéditeur du message SIP en modifiant l'identité de l'expéditeur d'un message [17].

- Vols de services : le pirate peut emprunter l'identité d'un utilisateur et l'utiliser pour faire passer des appels sur le réseau ToIP sans avoir payer le fournisseur de service [17].

4.3 Solutions de sécurité de la ToIP

Les mécanismes de sécurité proposés dans une architecture ToIP sont :

- La sécurité de l'infrastructure IP: C'est le premier niveau de sécurité, car la sécurité de l'infrastructure ToIP est liée à la sécurité du réseau IP. Un exemple est la séparation logique des réseaux Data et Voix par des VLANs.

- L'authentification : L'authentification du téléphone IP par le serveur et l'authentification du serveur par le téléphone IP avant d'autoriser un quelconque appel. Il existe différents moyens d'authentification tels que: SIPS, IPsec.

  - ☑ SIPS (Session Initiation Protocol Secure) : est un mécanisme de sécurité défini par RFC 3261 pour l'envoi de messages SIP au dessus du protocole de sécurisation TLS (Transport Layer Security).

  - ☑ IPsec (Internet Protocol Security) : est un ensemble de protocoles (couche 3 du modèle OSI) défini par IETF (RFC 2401), permettant le transport sécurisé des données sur un réseau IP [6].

- Le chiffrement: c'est un moyen efficace de protéger les données. Plusieurs solutions peuvent être utilisées : le chiffrement des flux de signalisation avec SIPS, le chiffrement des flux voix avec SRTP, des solutions propriétaires.

  - ☑ SRTP (Secure Real-time Transport Protocol): définit un profil de RTP, qui a pour but d'apporter le chiffrement, l'authentification et l'intégrité des messages, et la protection contre le replay de données RTP en unicast et multicast. SRTP a été conçu par Cisco et Ericsson, et est ratifié par l'IETF en tant que RFC 3711.

5. La ToIP et IPv6

IPv6 est le protocole Internet de nouvelle génération conçu par l'IETF. Il permet principalement de disposer d'un plus grand nombre d'adresses pour chaque élément du réseau. Il offre également une plus grande facilité de configuration et améliore les mécanismes de gestion de la mobilité IP. Il intègre nativement la sécurité, les classes de service et la diffusion multicast.

Le déploiement du protocole IPv6 pour le support de la ToIP va permettre d'éviter d'avoir recours aux NATs grâce à sa grande capacité d'adressage et de simplifier ainsi l'architecture. Néanmoins, la mise en place de la ToIP en IPv6 n'est pas encore répandue parce que la plupart des équipements de ToIP ne supportent pas encore cette version du protocole IP. Voici des exemples de solutions ToIP qui supportent IPv6: Kamailio (routeur d'appels), Linphone, Kphone et SJPhone (qui sont des « softphones », applications logicielles à installer sur un ordinateur).

Conclusion

La téléphonie sur IP est une technologie qui utilise les réseaux informatiques comme support de communication. Les solutions ToIP sont de plus en plus basées sur des standards ouverts. Beaucoup de ces solutions utilisent SIP comme protocole de signalisation ToIP. Les principaux protocoles utilisés pour le transport de la voix sont : RTP et RTCP. Et pour garantir la compatibilité entre la ToIP et le réseau téléphonique classique, l'IETF a travaillé sur le standard ENUM.

Le déploiement de la technologie ToIP dans les réseaux actuels a provoqué l'apparition des nouvelles problématiques notamment au niveau des dispositifs de sécurité tels que les pare-feu et les routeurs NAT, ainsi que les vulnérabilités de ToIP en terme de sécurité. Les problèmes de NAT et de pare-feux ont été solutionnés en utilisant plusieurs techniques qui sont résumées dans le tableau 2, tandis que de nombreux mécanismes de sécurité ont été présentés notamment la sécurité de l'infrastructure IP, l'authentification, et le chiffrement.

Dans le chapitre qui suit, je vais présenter la maquette expérimentale ToIP qui est la plate-forme de travail que j'ai utilisée initialement pendant mon stage. Par la suite, je vais présenter le travail effectué pendant le stage.

Chapitre 3

CHAPITRE III : Design et ingénierie ToIP

Introduction

La démarche expérimentale est la méthode qui caractérise le travail de l'équipe SIPA pour la mise en production des nouveaux services au profit de la communauté RENATER.

Pour la mise en production d'un service de routage d'appels, une maquette expérimentale d'interconnexion de sites universitaires et centres de recherche, ayant déployé une solution de ToIP, en interne, a été mise en place par l'équipe SIPA.

Aujourd'hui, la maquette expérimentale ToIP de RENATER présente des limites, notamment les aspects de supervision ne sont pas implémentés, les statistiques sur les appels traités par le serveur de routage d'appels OpenSER ne sont pas établies, et les accès au serveur ne sont pas sécurisés.

Dans les chapitres qui suivent, Nous commencerons tout d'abord par voir un aperçu sur la maquette expérimentale ToIP de RENATER, nous poursuivons ensuite par la présentation des réalisations pendant notre projet de fin études.

A. Présentation de l'existant

1. Description de la maquette ToIP de RENATER

1.1 Architecture de la maquette

La maquette expérimentale d'interconnexion des sites en ToIP repose sur le protocole SIP. Elle est composée par:

- Les IPBXs (Mitel, Cisco, Alcatel, Asterisk, etc.) mis en place aux niveaux des sites universitaires et les centres de recherche pour offrir le service de ToIP à leurs usagers.
- Un routeur d'appel IP qui est basé sur le routeur SIP OpenSER. Son rôle est d'assurer l'interconnexion des dits IPBXs, et le routage d'appels inter-site.

Cette maquette permet :

- De centraliser tous les préfixes atteignables au niveau du plan d'adressage et d'acheminer les appels inter-site sur le réseau RENATER.
- D'offrir une souplesse organisationnelle dans la gestion des préfixes pour les établissements.

Actuellement la maquette de ToIP permet de mettre en relation une dizaine de sites : l'INRIA (3000 postes), les Universités de Normandie via le CRIHAN (4770 postes), les Universités de Lorraine (5910 postes), l'université de Montpellier III (900 postes) ainsi que le GIP RENATER (50 postes répartis entre les sites de Paris et Montpellier).

La figure suivante illustre l'architecture de la maquette expérimentale ToIP de RENATER avec des exemples des sites raccordés à la maquette [16]:

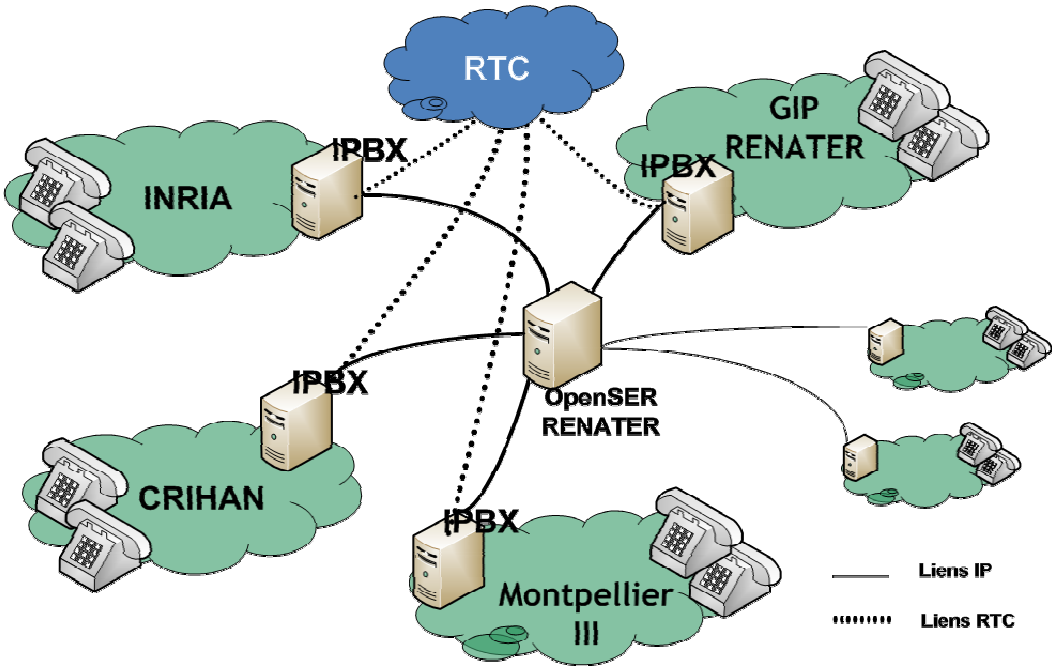

Figure 18 : maquette expérimentale ToIP de RENATER

1.2 Principe de fonctionnement

Le site souhaitant profiter du service ToIP afin d'acheminer ses appels à destination d'autres sites ayant-droit RENATER, n'aura besoin de paramétrer son IPBX qu'une seule fois. Au moins deux routes devront exister :

- Une route vers RENATER
- Une route vers l'opérateur

Le serveur de routage d'appels OpenSER assure la mise en relation entre les différents IPBX des sites connectés à la maquette.

L'utilisateur compose le numéro de son correspondant. Si le numéro n'est pas joignable en IP, le serveur OpenSER renvoie un message SIP, qui permet à l'IPBX du site origine de basculer l'appel sur la route suivante, le plus souvent son accès RTC (Réseau Téléphonique Commuté). L'intérêt principal de cette maquette est que le site n'aura pas besoin de

connaitre les routes pour joindre l'ensemble des IPBX de la communauté RENATER et n'aura donc pas à maintenir à jour une table de routes vers les sites.

Le schéma suivant décrit le principe de fonctionnement de la maquette.

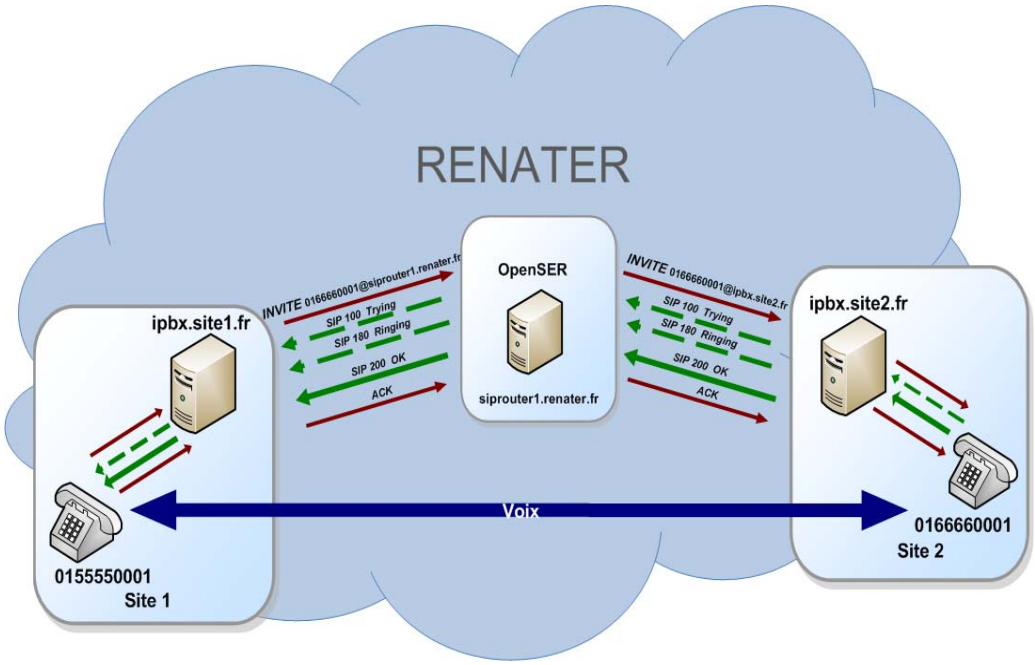

Figure 19 : principe de fonctionnement de la maquette

L'organigramme suivant montre le principe de fonctionnement d'OpenSER.

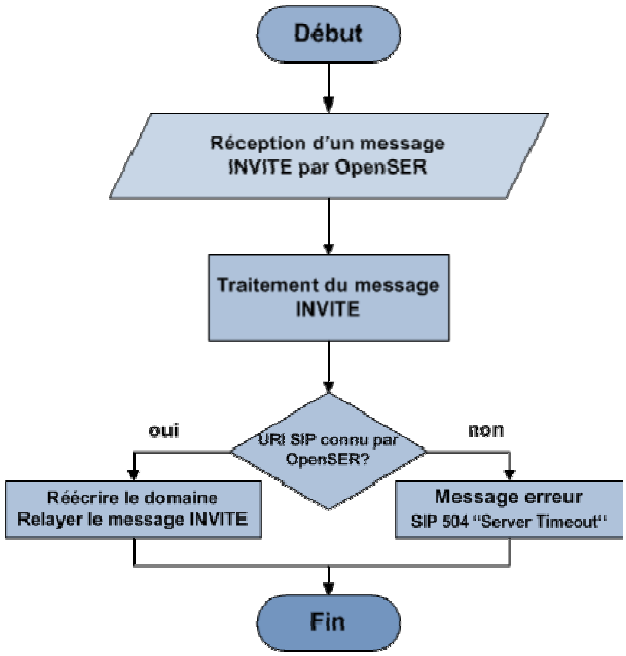

Figure 20 : principe de fonctionnement du serveur OpenSER

2. Présentation d'OpenSER

Le routeur d'appel OpenSER constitue l'élément central de la maquette. Chaque requête SIP INVITE d'établissement de communication émise par un IPBX est traitée par le proxy OpenSER.

OpenSER est un logiciel libre, Il est une version héritée de SER (SIP Express Router), le code (en licence GPL) de SER a été repris par un groupe de développeurs du projet à la fin de l'année 2005 pour constituer un nouveau logiciel.

OpenSER prend en charge les fonctions :

- Proxy server : il assure les fonctions de relayage des requêtes et réponses SIP entre deux *Users Agents*.
- Registrar server : il gère les requêtes *REGISTER* envoyées par les *Users Agents* pour signaler leur emplacement courant.
- Location server : il permet de fournir les détails d'emplacement courant d'un utilisateur.
- Redirect server : il redirige les *Users Agents* vers un autre Proxy server.
- Application server : il fournit des services avancés pour les utilisateurs tels que service de présence, messagerie instantanée, etc.

Voici ci-dessous les composantes d'OpenSER :

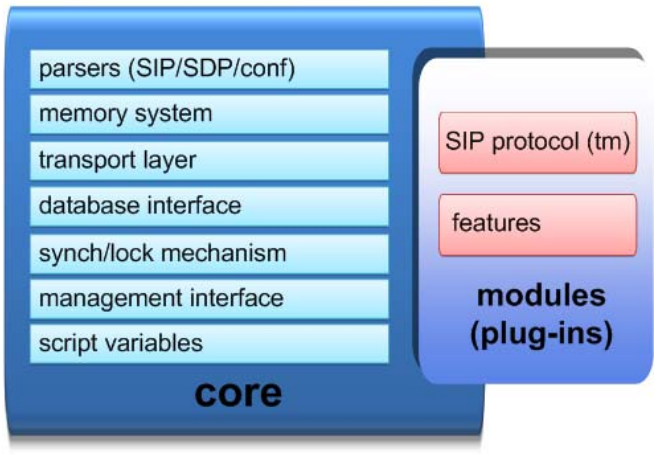

Figure 21 : composantes d'OpenSER

B. Travail effectué

1. Évolutions de la plate-forme de routage d'appels OpenSER

1.1 Objectifs

L'objectif est d'installer un nouveau routeur d'appels pour le service pilote en se basant sur les travaux effectués jusqu'à présent sur la maquette expérimentale de RENATER.

1.2 Etude

Dans cette étude, j'ai trouvé que des versions plus récentes que la version installée d'OpenSER sont disponibles. Ces nouvelles versions contiennent des améliorations des fonctionnalités, et des corrections de bugs. J'ai découvert que le projet OpenSER a changé son nom en 'Kamailio' [2] et qu'un nouveau projet appelé 'OpenSIPS' [3] a été lancé sur la base d'OpenSER. J'ai donc été amené à faire une comparaison des deux projets. La comparaison proposée est basée sur les fonctionnalités, la taille de la communauté travaillant à cette version, et la dynamique de chaque projet.

Voici un tableau qui résume les résultats de la comparaison effectuée le 20 septembre 2008.

Critère	Kamailio	OpenSIPS
résultats de recherche Google	Kamailio+SIP 25 400	OpenSIPS+SIP 16 600
Dernière mise jour du site	2008-10-02	2008-09-01
Nombre d'administrateur du projet	5	1
Nombre de développeurs	27	16
Licence d'utilisation	GPL	GPL
La version stable	Kamailio v1.4.1	OpenSIPS 1.4.2
La version en cours de développement	Kamailio v1.5.0	----------
Mailing lists	oui	Oui
Nombre de modules (fonctionnalités)	86	86
Communauté	active	Moins active

Tableau 3 : résumé des résultats de comparaison entre Kamailio et OpenSIPS

A la lecture des résultats de cette comparaison, notre choix s'est porté sur le projet Kamailio dans un premier temps. Cependant, nous suivons de près l'évolution du projet OpenSIPS.

1.3 Evolutions

La principale évolution est la mise en place d'un service pilote de routage d'appels basé sur Kamailio pour la communauté RENATER, la version installée est Kamailio 1.4.1 (cf. ANNEXE – 1).

La maquette expérimentale restera en place afin de permettre à des nouveaux sites de faire des tests avant de se connecter au pilote. Elle permettra également de valider des nouveaux services avant de les mettre en production sur le pilote (IPv6, redondance, etc.).

La migration des sites raccordés à la maquette vers le service pilote se fait après la validation du fonctionnement de la ToIP du site sur la maquette expérimentale.

Des améliorations ont aussi été introduites dans le fichier de configuration par rapport à la configuration d'OpenSER existante dans la maquette expérimentale notamment la prise en charge les messages SIP OPTIONS.

2. Mise en place d'une solution de supervision

2.1 Objectifs

L'objectif principal était de proposer une solution de supervision qui permette de :

- Surveiller l'état du service de routage des appels téléphoniques dans le serveur Kamailio de RENATER et ses performances (la charge CPU, utilisation de la mémoire, utilisation du disque dur).
- Superviser l'état des IPBX des sites distants interconnectés au service pilote.
- Grapher dans le temps l'état du serveur Kamailio et des IPBX des sites distants.
- Avertir l'administrateur du pilote ToIP en cas de problème.

2.2 Etude

La plupart des solutions de supervision de ToIP qui existent sont des solutions commerciales. Parmi les logiciels libres permettant de superviser les services voix, Monit [6] permet de surveiller des services locaux installés sur une machine Linux/Unix. On peut utiliser cet outil pour superviser le routeur SIP (Kamailio) de RENATER et les sites distants qui utilisent des logiciels IPBX (Kamailio,OpenSIPS, Asterisk ..), mais cette solution ne permet pas de superviser les sites distants qui utilisent des IPBX matériels comme (MITEL, CISCO …) et ne répond pas à tous nos besoins.

Une autre solution est d'utiliser l'outil SIPp qui permet de générer des appels SIP avec un UAC[4] et un UAS[5] en ligne de commande en exploitant le rapport d'appels élaboré par cet outil. Cette solution était trop limitée par rapport à l'ampleur de nos besoins car d'autres outils sont nécessaires pour vérifier l'état des liens et pour générer des graphiques. Cependant, l'outil SIPp pourra être utile pour tester les performances des communications entre deux sites.

La dernière solution étudiée est le logiciel de supervision Nagios [4], qui semble répondre très bien à nos besoins grâce à sa modularité et les greffons disponibles pour SIP. Le choix s'est donc porté naturellement sur Nagios.

2.3 Solution Nagios

Nagios est un logiciel libre de supervision de réseau. Il est disponible sous Licence GNU GPL, il permet de savoir à tout moment le statut des hôtes et des services spécifiés par l'administrateur réseau. Il se charge également de l'envoi d'alertes suite à une panne ou un dysfonctionnement du réseau.

Nagios s'appuie sur un moteur écrit en C chargé de l'ordonnancement des vérifications, ainsi que les actions à réaliser lors de la détection d'un incident. Les vérifications sont faites à l'aide des greffons (plugins) qui permettent aux utilisateurs de développer facilement leurs propres vérifications des services. Les greffons peuvent être développés dans n'importe quel langage de programmation (C, shell, perl, …). Le tout est contrôlable à travers une page web en CGI et accessible via n'importe quel serveur HTTP comme Apache.

Voici un schéma détaillant l'architecture de base de Nagios :

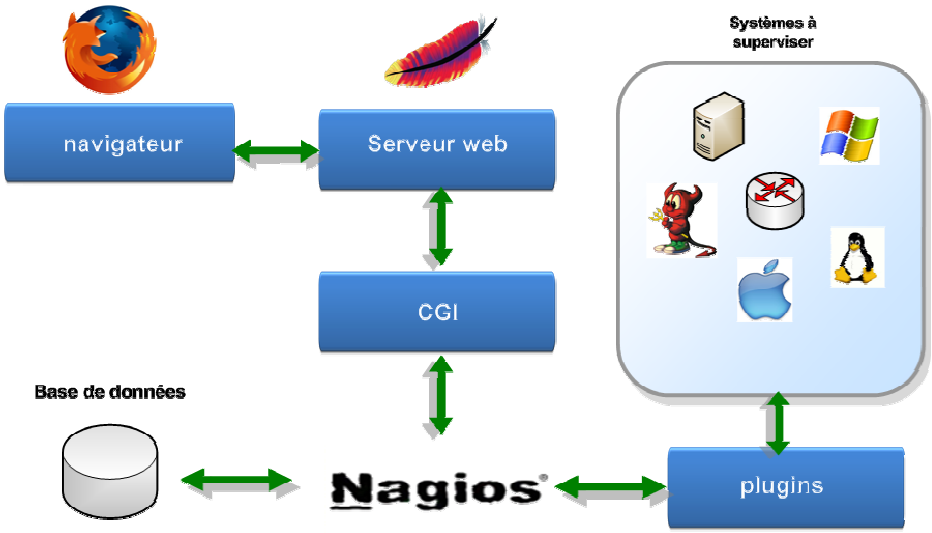

Figure 22 : architecture de base de Nagios

[4] Il initie la session
[5] Il répond aux requêtes

L'installation de Nagios est réalisée sur la distribution Debian Etch avec la commande apt-get de Linux, qui permet de récupérer les fichiers de Nagios et de procéder facilement à l'installation et à la configuration de ce dernier. La version installée est Nagios 3.0.6. Nous avons aussi besoin d'installer les plugins Nagios qui sont utilisés pour la supervision. La version installée est nagios-plugins 1.4.13.

Pour superviser les performances de la machine Kamailio, j'ai utilisé le plugin NRPE qui permet d'exécuter des plugins à distance sur la machine Kamailio puis d'envoyer les résultats à Nagios (cf. ANNEXE - 2).

Pour surveiller le service de routage des appels sur le routeur SIP (Kamailio) de RENATER et les IPBX des sites distants, j'ai utilisé le plugin check_sip qui n'est pas fourni en standard avec les plugins de Nagios.

Pour tracer des graphiques pour les différents services supervisés, j'ai installé le module PNP (l'acronyme de PNP est *PNP is NOT Perfparse*). Ce module permet de récupérer les données relatives aux performances du système interrogé et d'injecter ces valeurs dans des bases rrdtool. Il est possible ainsi de générer des graphiques personnalisés intégrés à l'interface web.

J'ai configuré Nagios pour qu'il avertisse l'administrateur du pilote ToIP par e-mail en cas de problème.

Enfin j'ai mis en place le style Nuvola pour améliorer l'interface graphique Nagios et la rendre plus agréable.

Les figures ci-dessous présentent une vue d'ensemble des interfaces de la solution Nagios mise en place :

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
GIP RENATER PARIS	PING	OK	2009-01-20 13:37:58	6d 21h 6m 17s	1/4	PING OK - Paquets perdus = 0%, RTA = 0.76 ms
	SIP	OK	2009-01-20 13:41:18	0d 0h 36m 17s	1/4	SIP/2.0 200 OK, 0.076474 seconds response time
INSERM	PING	OK	2009-01-20 13:39:36	0d 1h 47m 59s	1/4	PING OK - Paquets perdus = 16%, RTA = 37.89 ms
	SIP	CRITICAL	2009-01-20 13:37:33	0d 2h 33m 2s	4/4	Unknown error: SIP/2.0 504 Serveur Timeout, 0.057977 seconds response time
Kamailio	CPU Load	OK	2009-01-20 13:40:33	1d 2h 17m 2s	1/4	OK - Charge moyenne: 0.00, 0.00, 0.00
	Disk Space	OK	2009-01-20 13:40:53	1d 2h 21m 42s	1/4	DISK OK - free space: / 2534 MB (94% inode=97%): /lib/init/rw 1643 MB (100% inode=99%): /dev 9 MB (99% 1643 MB (100% inode=99%): /boot 2599 MB (97% inode=99%): /home 35911 MB (99% inode=99%): /o inode=99%): /tmp 876 MB (98% inode=99%): /usr 8261 MB (92% inode=98%): /var 8478 MB (95%
	PING	OK	2009-01-20 13:37:40	4d 20h 31m 46s	1/4	PING OK - Paquets perdus = 0%, RTA = 0.46 ms
	SIP	OK	2009-01-20 13:40:27	0d 0h 37m 8s	1/4	SIP/2.0 200 ok, 0.151743 seconds response time
	memory Load	CRITICAL	2009-01-20 13:40:33	1d 1h 32m 2s	4/4	Memory CRITICAL - 6.0% (166596 kB) free

Figure 23 : interface de l'état des services supervisés

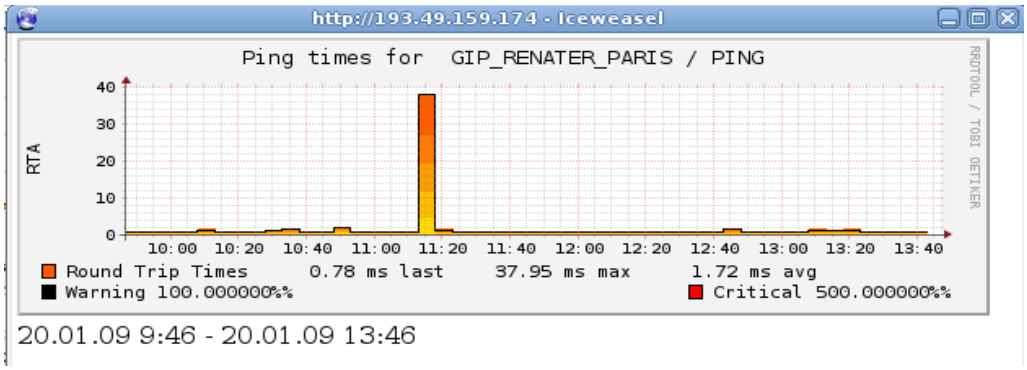

Figure 24 : interface du temps de réponse du service Ping

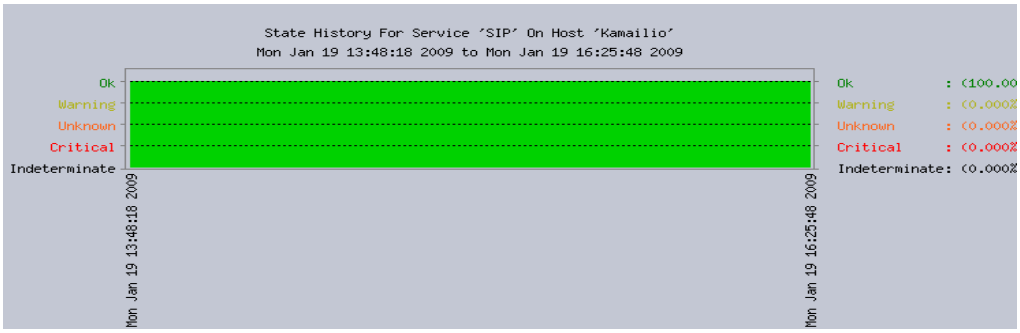

Figure 25 : interface de l'état du service SIP de serveur Kamailio dans le temps

Durant la mise en place de toutes ces solutions, j'ai rencontré plusieurs problèmes de configuration à cause du manque de documentation. Un problème est posé par le module PNP lors de l'affichage des graphiques du service SIP. En effet, PNP utilise des modèles pour générer les graphiques. Le modèle (template) du service SIP n'est pas défini, j'ai donc dû écrire notre propre modèle.

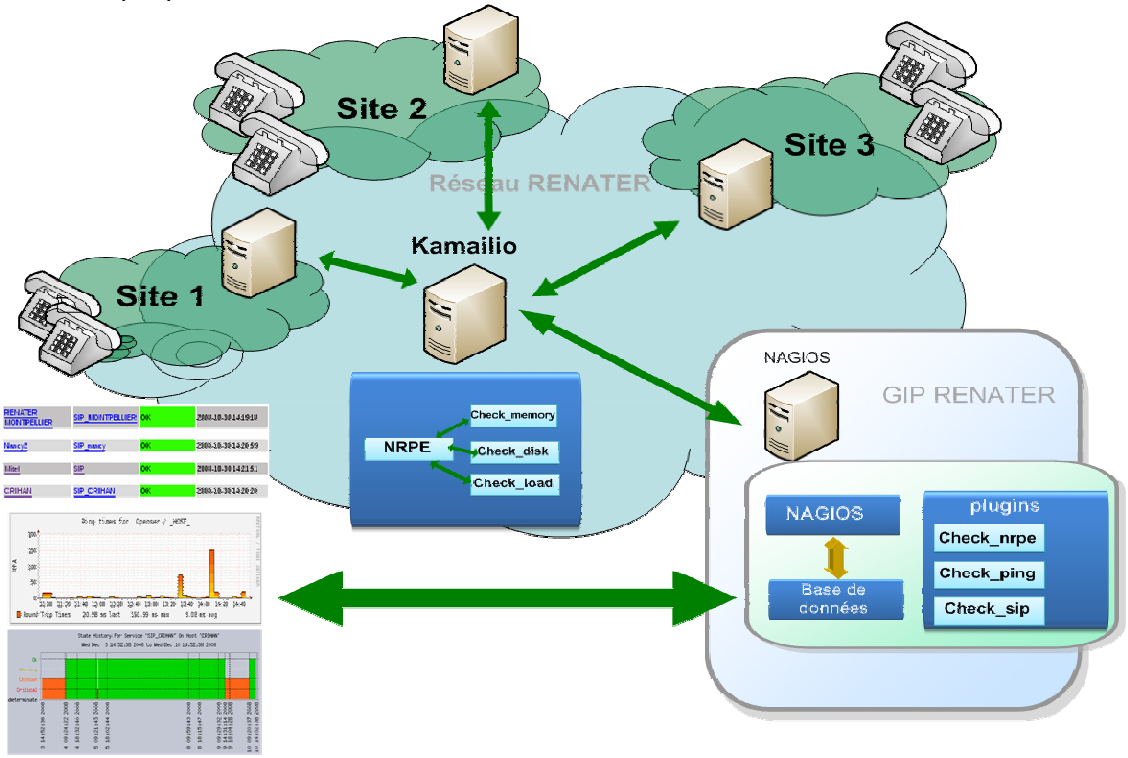

Figure 26 : architecture de la solution mise en place pour la supervision

3. Mise en place d'une solution de Comptabilisation d'appels

Le but de la mise en place d'une solution de comptabilisation d'appels est de pouvoir fournir des statistiques sur le taux d'utilisation du service de routage d'appels de RENATER et de collecter les informations à analyser en cas de problème.

3.1 Objectifs

L'objet de cette partie de mon stage est d'étudier et mettre en place une solution de comptabilisation d'appels qui permette de :

- Etablir des statistiques sur les appels traités par le serveur Kamailio de RENATER.
- Suivre ces statistiques dans le temps.
- Agréger les statistiques par site pour avoir un suivi précis de l'usage du service pour chaque site.

3.2 Etude

Après avoir effectué plusieurs recherches sur les solutions de comptabilisation d'appels existantes qui peuvent répondre à nos besoins, j'ai trouvé qui il y a de nombreuses solutions qui sont soit commercialisées, soit trop complexes pour nos besoins. Notre étude est consacrée à trois solutions.

La première est d'utiliser les données du fichier log du serveur kamailio, cependant cette solution ne permet pas d'avoir un niveau de détails suffisant sur l'appel.

Une deuxième solution consiste à utiliser le module ACC[6] avec une base de données MySQL. L'inconvénient de cette solution est la génération d'un enregistrement dans la base de données pour chaque requête SIP reçue par le serveur Kamailio [7].

Une troisième solution étudiée est d'utiliser le module ACC avec un serveur RADIUS. C'est cette dernière solution qui a été adoptée. En effet, elle va permettre d'avoir un seul enregistrement pour toute la session SIP.

3.3 Mise en place

RADIUS est un protocole qui intègre les notions d'AAA (Authorization, Authentication and Accounting). La dernière version du protocole RADIUS est normalisée par l'IETF dans deux RFC : RFC 2865 (RADIUS authentication) et RFC 2866 (RADIUS accounting).

[6]ACC est un module de Kamailio qui permet d'enregistrer les transactions SIP dans différentes bases de donnés

Nous avons choisi FreeRADIUS [5] comme serveur RADIUS pour la fonction d'*accounting*. Le principe de cette fonction se base sur deux types de paquets principaux: *Accounting Start* et *Accounting Stop*. Une session est définie par l'intervalle entre un *Start* et un *Stop*.

FreeRADIUS est un logiciel libre, il s'agit d'un des serveurs RADIUS les plus modulaires et riches en fonctionnalités. Les détails de l'installation et la configuration sont expliqués en annexe 3 de ce rapport.

Dans cette solution, le serveur Kamailio va émettre un paquet *Accounting Start* avec un identificateur de session au serveur FreeRADIUS lors de la réception d'un message *SIP INVITE*. Quand le serveur Kamailio reçoit le message *SIP BYE* de la même session, il envoie un paquet *Accounting Stop* avec le même identificateur de session. Le serveur freeRADIUS va envoyer alors un seul enregistrement à la base de données MySQL qui contient la durée et tous les paramètres de la session.

Voici un schéma décrivant le principe de fonctionnement de cette solution :

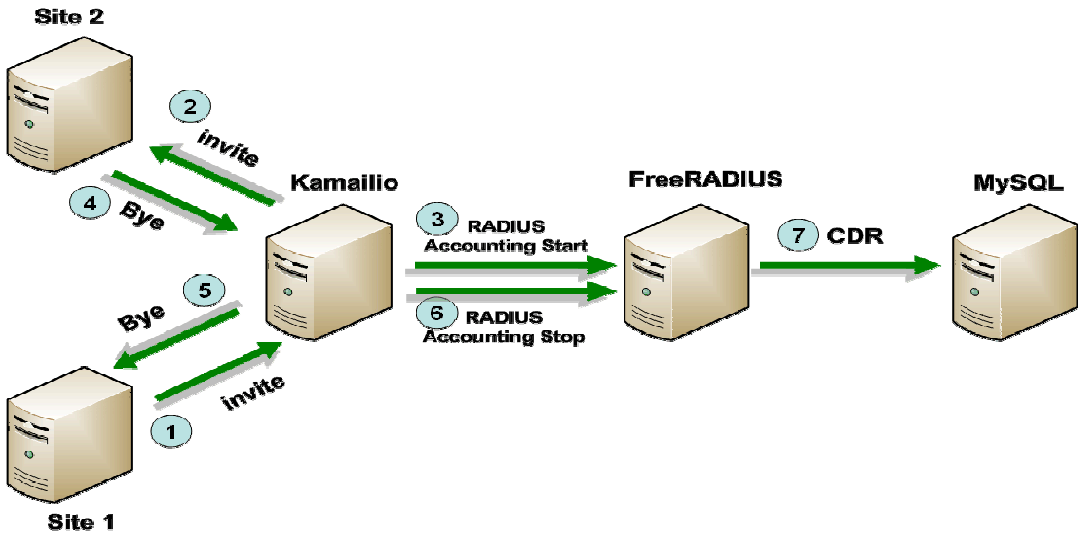

Figure 27 : principe de fonctionnement de FreeRADIUS avec Kamailio

Lors des tests effectués pour valider cette solution, nous avons trouvé que FreeRADIUS ne permet pas de générer les CDR pour les appels échoués, ce qui est normal, étant donné que le message *SIP BYE* n'est pas reçu par Kamailio. Les sessions n'ayant pas reçues de message SIP BYE sont comptabilisées dans un fichier log FreeRADIUS. J'ai donc été amené à mettre en place la solution du module ACC avec MySQL pour comptabiliser juste les appels échoués dans la base de données (cf. ANNEXE – 4).

Voici un schéma illustrant le principe de fonctionnement de la solution module ACC avec MySQL :

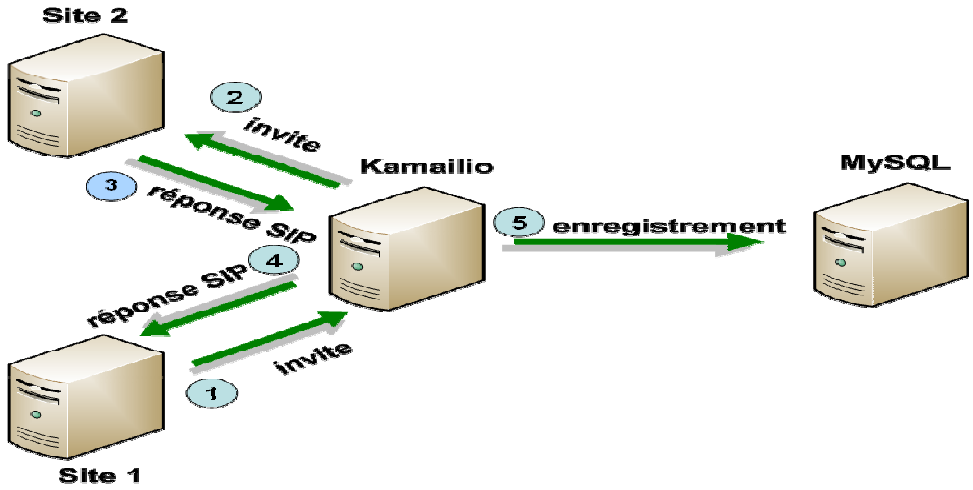

Figure 28 : principe de fonctionnement de module ACC avec MySQL

La solution de comptabilisation d'appels se compose de deux briques fonctionnelles :

1. Module ACC avec FreeRADIUS
2. Module ACC avec MySQL

En effet, la génération des CDRs relatifs à une communication SIP dépend essentiellement de la réussite de son traitement (existence ou non de messages d'erreur).

L'organigramme qui suit résume l'essentiel de ce qui a précédé.

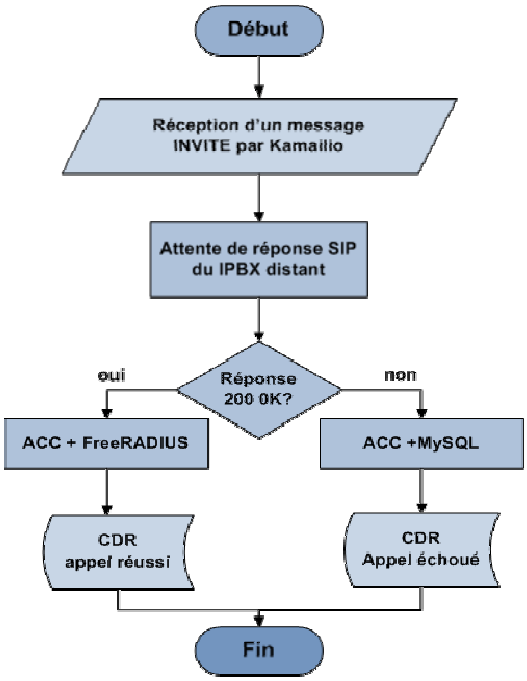

Figure 29 : organigramme de la solution de comptabilisation d'appels

La figure ci-dessous illustre l'architecture de la solution de comptabilisation d'appels :

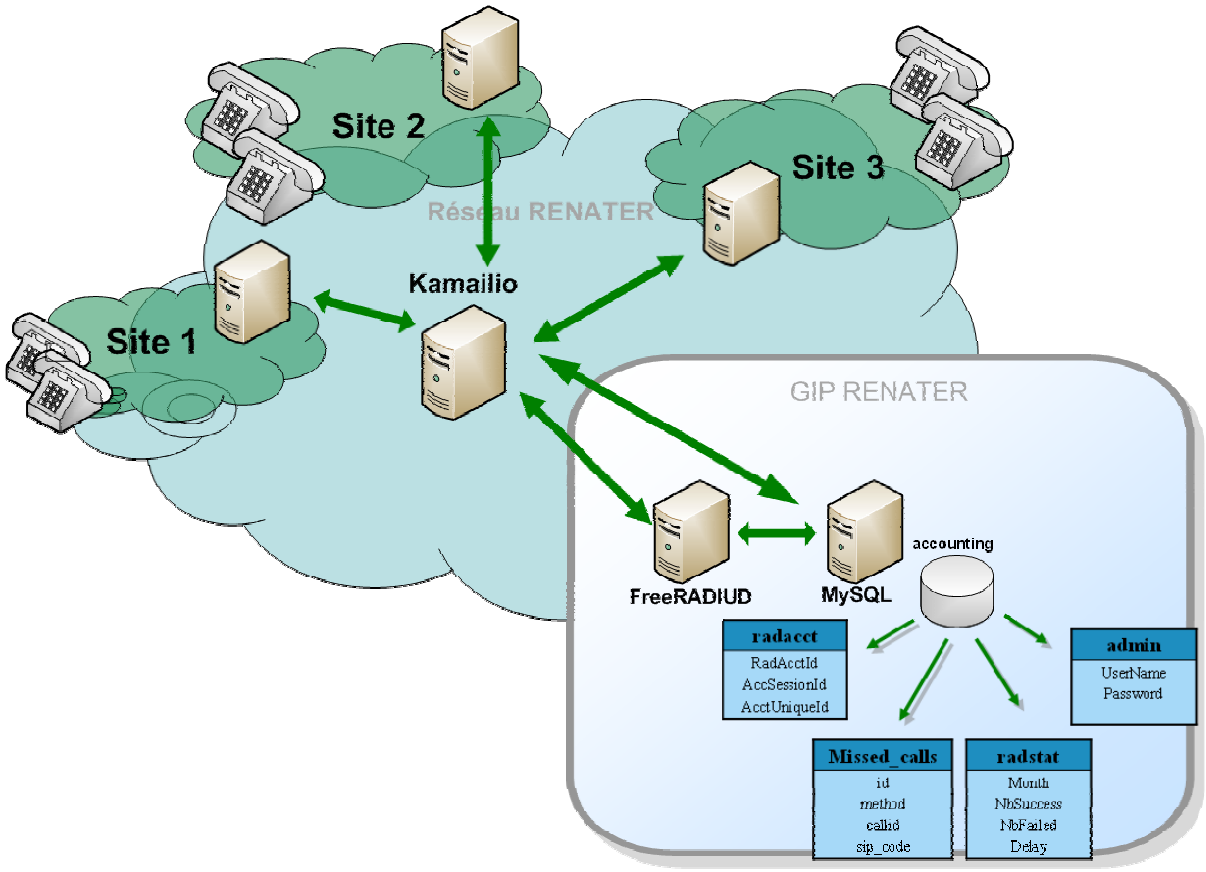

Figure 30 : architecture de la solution de comptabilisation des appels

3.4 Développement d'une interface web

Pour afficher les informations des appels stockées dans la base de données par FreeRADIUS, j'ai développé une interface web en PHP qui permet :

- Afficher le nombre des appels effectués par chaque site (figure 24)
- Afficher les détails des appels (figure 25)
- Afficher sous forme de graphes le nombre et la durée des appels effectués par mois et par année (figure 26)
- Effectuer des recherches sur les appels selon des critères(le site appelant, le site appelé, la date d'appel) (figure 27)

Voici les principaux écrans de l'application web développées :

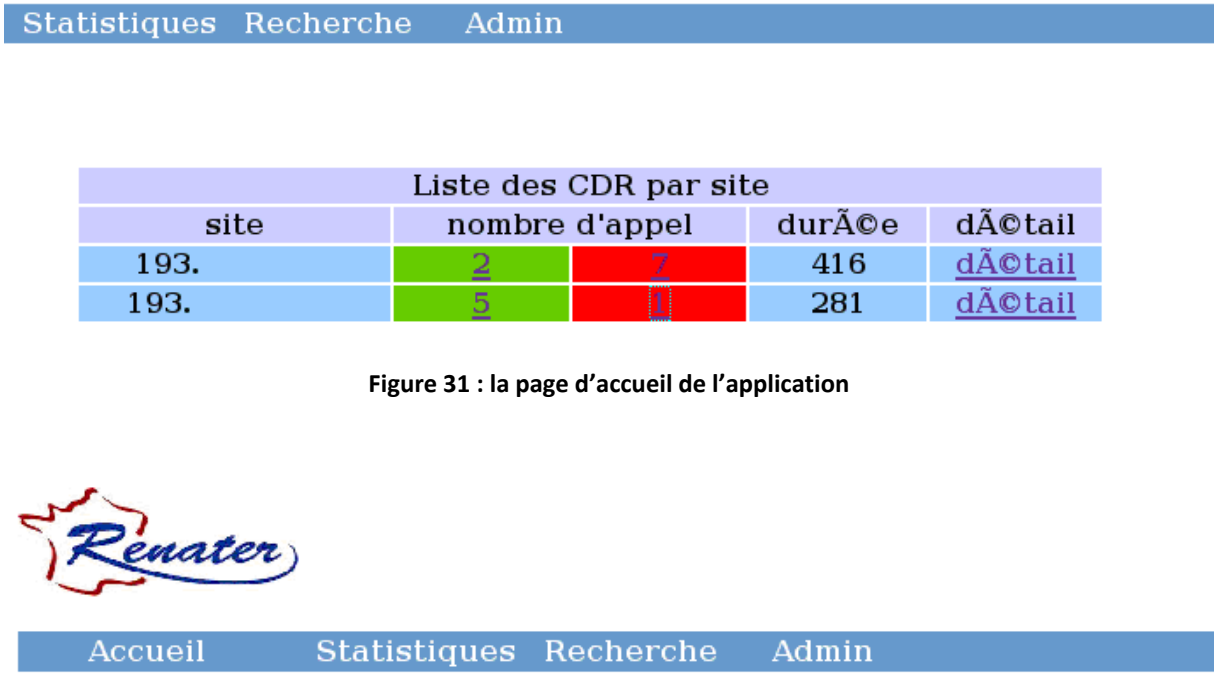

Figure 31 : la page d'accueil de l'application

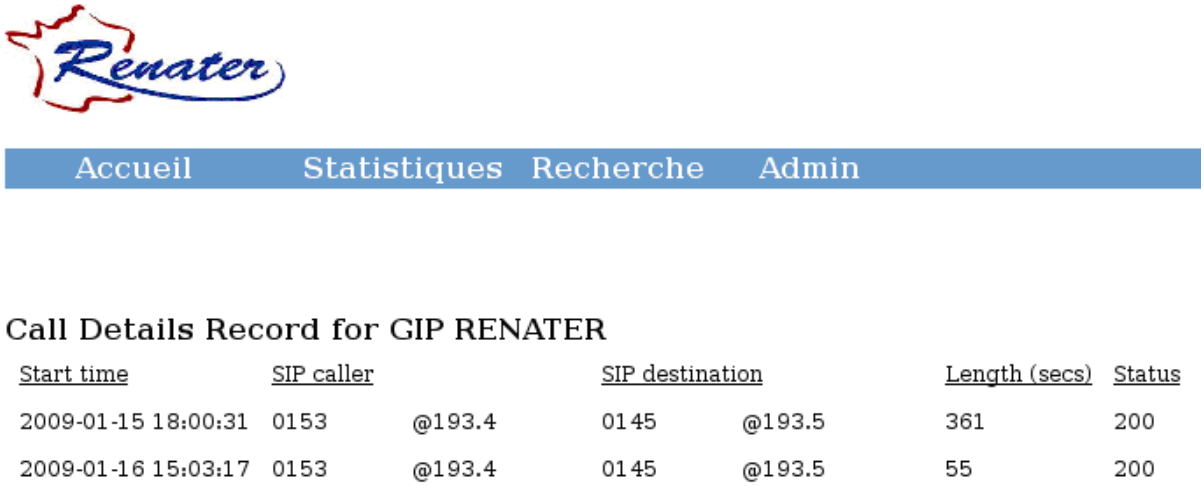

Figure 32 : ecran des détails des appels

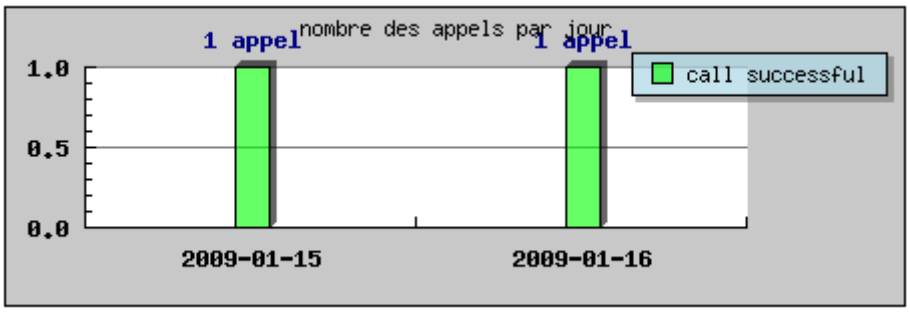

Figure 33 : ecran de nombre des appels par jour

Figure 34 : ecran de rechercher sur les appels

3. Sécurisation du pilote ToIP

Tout système accessible via IP est une cible potentielle pour l'ensemble des menaces pesant sur les réseaux IP. Afin de limiter les accès et sécuriser le serveur pilote ToIP, j'ai mis en place solution de sécurité basée sur l'outil IPtables. Des filtres ont été configurés sur le routeur d'appels de façon à n'autoriser que les IPBX des sites connectés au pilote à envoyer des requêtes SIP sur le port 5060. Les flux du serveur de supervision Nagios et de comptabilisation d'appel FreeRADIUS, ainsi que les flux pour l'administration du serveur à distance (SSH) sont autorisés également (cf. ANNEXE – 5).

Les règles de filtrage s'appliqueront donc selon les critères suivants :

- Adresse IP source
- Adresse IP de destination
- Protocole de communication
- Port source
- Port de destination

L'organigramme suivant résume la solution pour sécuriser le service pilote ToIP.

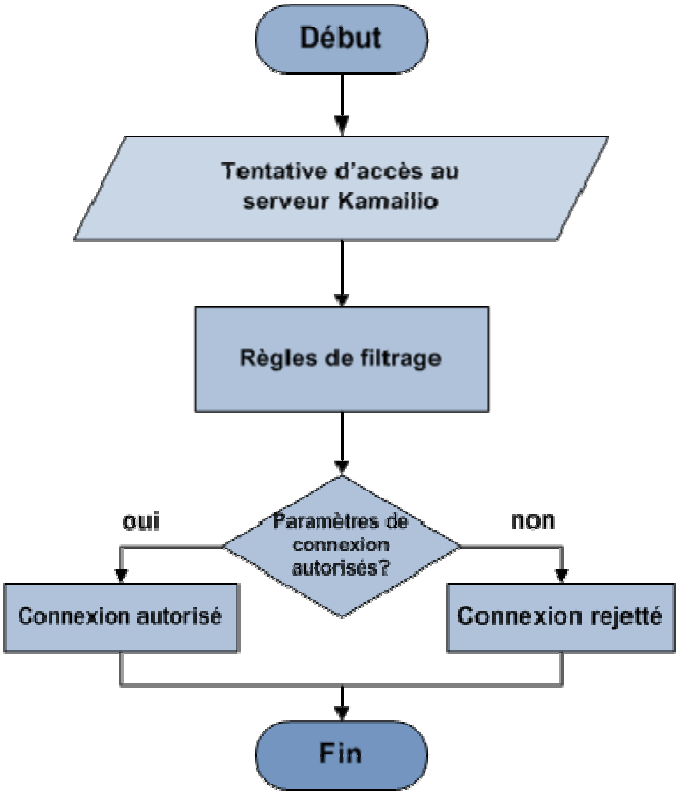

Figure 35 : organigramme de la solution de sécurisation du service pilote

4. Gestion de l'accessibilité des sites

L'objectif était de gérer les messages d'erreur SIP envoyés par le serveur de routage d'appels Kamailio, afin d'avoir un basculement rapide vers le réseau RTC en cas d'inaccessibilité du site distant. Pour cela, des tests ont été réalisés avec un des sites de l'INSERM (Institut National de la Santé et de la Recherche Médicale).

Le tableau suivant décrit les résultats des tests après les améliorations introduites dans le fichier de configuration de Kamailio.

Tableau 4 : les résultats des tests de la gestion des messages d'erreur

5. Tests de validation

Afin de valider les différentes solutions mises en place, des tests de validation ont été effectués avec le site INSERM.

Le tableau ci dessous donne les résultats des tests.

N° test	Description	Type de message erreur	Résultat de basculement	Temps de basculement
1	Désactiver l'interface réseau de l'IPBX	Requeste timeout	OK	10s
2	Arrêter le service SIP de l'IPBX	Requeste timeout	OK	10s

La solution	N° test	Description	Résultat attendu	Résultat
La solution de supervision	1	Désactiver l'interface réseau de l'IPBX	Nagios retourne un état de connectivité différent de OK	OK
	2	Arrêter le service SIP de l'IPBX	Nagios retourne un état du service SIP différent OK	OK
La solution de Comptabilisation d'appels	1	Effectuer un appel réussi	Enregistrement ajouté dans la base de données comme appel réussi	OK
	2	Effectuer un appel échoué (le service n'est pas disponible)	Enregistrement ajouté dans la base de données comme appel échoué	OK
La sécurisation du pilote ToIP	1	Effectuer un appel via l'OpenSER de la maquette	Appel rejeté par le routeur d'appels du service pilote	OK

Tableau 5 : les résultats des tests de validation

Conclusion

Dans ce dernier chapitre, nous avons présenté la maquette expérimentale, la phase de réalisation de notre projet en présentant les solutions mises en place, la démarche de travail, le principe de fonctionnement de chaque solution, finalement, nous avons présenté les tests de validation effectués.

Actuellement le service pilote ToIP est opérationnel, deux sites ont déjà migré vers ce service qui sont GIP RENATER PARIS et GIP RENATER MONTPELLIER. Il est prévu aussi de migrer d'autres sites dans les jours qui viennent.

Conclusion générale

La téléphonie sur IP constitue incontestablement une attraction de taille à la fois pour les équipementiers, les opérateurs, les entreprises et le grand public. Si les enjeux économiques justifient largement cette convoitise, il ne faut cependant pas négliger les contraintes techniques à surmonter.

Durant le présent projet de fin d'études, Il nous a été confié la mission, au sein de l'équipe SIPA (Services IP Avancés et prospective) du GIP RENATER, d'étudier et mettre en place le service pilote ToIP de RENATER. Pour cela, notre travail a été décomposé en cinq étapes majeures. La première avait pour but d'étudier les bases de la téléphonie sur IP. Le second travail consistait à étudier les évolutions possibles sur la maquette expérimentale mise en place par l'équipe SIPA pour installer le service pilote ToIP. La troisième étape était la mise en place d'une solution de supervision du service pilote. La quatrième étape consistait à mettre en place d'une solution de comptabilisation d'appels. Enfin, la dernière étape avait pour but de limiter les accès et sécuriser le serveur pilote ToIP. Ce projet de fin d'études a donné lieu à une plate forme de ToIP basée sur un serveur de routage d'appel de type Kamailio et des solutions pour la supervision et la comptabilisation des sessions SIP VoIP.

L'élaboration de ce travail m'a permis, d'une part, d'approfondir les connaissances et le savoir faire acquis durant les années de ma formation à l'ENSA de Marrakech, et d'autre part, de préparer mon intégration à la vie professionnelle et de me se situer sur le marché des télécommunications (réseaux, systèmes de communication, services, ...).

Le travail que j'ai réalisé pourrait être complété et poursuivi sous différents aspects, notamment :

- Mise en place d'une solution de redondance des serveurs impliqués dans le service pilote.
- Introduction d'IPv6 dans les équipements du service pilote et avec les sites partenaires.
- Amélioration des mécanismes de sécurité mis en place.

Références bibliographiques

[1]. Site officiel de RENATER, **http://www.renater.fr,** janvier 2009

[2]. Site officiel de Kamailio**,** **http://www.kamailio.org,** janvier 2009

[3]. Site officiel d'OpenSIPS**,** **http://www.opensips.org,** janvier 2009

[4]. Site officiel de Nagios**,** **http://www.nagios.org,** janvier 2009

[5]. Site officiel de FreeRADIUS, **http://freeradius.org,** janvier 2009

[6]. L'encyclopédie libre wikipedia**,** **http://fr.wikipedia.org/wiki/Accueil,** janvier 2009

[7]. Flavio E. Goncalves, **Building Telephony Systems with OpenSER,** Packt Publishing (April 25, 2008)

[8]. N. IDBOUFKER, **Architectures Protocolaires VoIP : H.323 et SIP**, décembre 2008

[9]. R. Sparks, M. Handley, E. Schooler, **'SIP: Session Initiation Protocol'**, RFC 3261, IETF, June 2002

[10]. F. Andreasen, B. Foster, **'Media Gateway Control Protocol (MGCP) '**, RFC 3435, IETF, January 2003

[11]. P. Faltstrom, M. Mealling, '**The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM) '**, RFC 3761, IETF, April 2004

[12]. K. Egevang, P. Francis, **' The IP Network Address Translator (NAT) ',** RFC 1631, IETF, May 1994

[13]. J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, **'STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)'**, RFC 3489, IETF, March 2003

[14]. C. Rigney, S. Willens, A. Rubens, W. Simpson, **'Remote Authentication Dial In User Service (RADIUS)'**, RFC 2865 , IETF, June 2000

[15]. C. Rigney, **'RADIUS Accounting'**, RFC 2866 , IETF, June 2000

[16]. Rapport du groupe de travail ToIP, **www.renater.fr/IMG/pdf/Compil_Doc_synth.pdf**

[17]. Best Practices for VoIP-SIP Security, **www.td.unige.ch/pdf/BP_VoIP_Security.pdf**

ANNEXES

ANNEXE – 1 : Configuration Kamailio

Ceci est un extrait du fichier de configuration du serveur Kamailio :

```
####### Rapport des modifications ##########
# création 16/12/2008
# Raccordement SITE1 (Ville1 et Ville2) le 17/12/2008
# test avec SITE2le 24/12/2008
####### Global Parameters #########
debug=3
log_stderror=no
log_facility=LOG_LOCAL0
fork=yes
children=4
port=5060
listen = udp:193.*.*.*
alias= *.renater.fr
……..
####### Modules Section ########
#set module path
mpath="//usr/local/lib/kamailio/modules/"
……..
########## Bloc des sites raccordés au service pilote #######
######################## GIP RENATER  ##################
……..
     if (uri=~"sip:01539420[3,4,8,9][0-9]@.*") {
     #         40       Postes
                 route(2);
    };
……..
# ------------ process traffic from Internet to SITE1 ----------------------
route[2] {
#    log(1, "In route[2]");
     rewritehostport("193.*.*.*:5060");
     append_hf("P-hint: Forwarded to SITE1\r\n");
     xlog("L_INFO", "$rm from $fu to $tu");
     if (!t_relay()) {
           sl_reply_error();
     };
exit;
}
```

ANNEXE – 2 : exemple de définition des services pour la supervision

Un exemple de définition des services à superviser pour la machine Kamailio :

```
define host{
      use                generic-host
      host_name          Kamailio
      address            193.*.*.*
}

define service{

 host_name              Kamailio
 service_description    SIP
 check_command          check_sip!sip:193.*.*.*
 use                    generic-service
 action_url             http://193.*.*.*/pnp4nagios/index.php?host=Kamailio&srv=SIP

}

define service {
 host_name              Kamailio
 service_description    PING
 check_command          check_ping!100.0,20%!500.0,60%
 use                    generic-service
 action_url             http://193.*.*.*/pnp4nagios/index.php?host=Kamailio&srv=PING
}

define service{
      use                           generic-service
      host_name                     Kamailio
      service_description           Disk Space
      check_command                 check_nrpe!check_hda1
      }

#Charge CPU
define service{
     use                            generic-service
     host_name                      Kamailio
     service_description    CPU Load
     check_command                  check_nrpe!check_load
}
```

ANNEXE – 3 : Configuration module ACC avec FreeRADIUS

Installation FreeRADIUS

- La version installée 2.0.* :

```
#apt-get install freeradius freeradius-utils
# apt-get install freeradius-mysql
```

Création et la configuration de la base de donnée de freeRADIUS

- Installation Mysql

```
# apt-get install mysql-server
# apt-get install mysql-client
```

- Création de la base de données

```
dpkg - i cdrtool_6.6.10_all.deb
mysqladmin -u root –p create accounting
mysql -u root accounting </var/www/CDRTool/setup/radius/OpenSIPs/radacct.mysql
```

Configuration du serveur FreeRADIUS

- La configuration de la connexion à la base de données

```
sql {
driver = "rlm_sql_mysql"
server = "127.0.0.1"
login = "root"
password = "**********"
radius_db = "accounting"
acct_table = "radacct"
sqltrace = no
sqltracefile = ${logdir}/sqltrace-%Y%m%d.log
num_sql_socks = 25
connect_failure_retry_delay = 60
```

- Configuration du serveur kamailio comme client de FreeRDIUS dans le fichier /etc/freeradius/clients.conf

```
client 193.*.*.* {
secret=**********
shortname= siprouter1
nastype=other
}
```

- Activer mysql dans le fichier /etc/freeradius/radiusd.conf

```
Accounting {
acct_unique
detail
sql
unix
radutmp
}
```

- Ajouter le dictionnaire kamailio dans le fichier /etc/freeradius/dictionary

```
cp /var/www/CDRTool/setup/radius/OpenSIPs/dictionary.ser
/etc/freeradius/dicotionary.kamailio

$INCLUDE /usr/share/freeradius/dictionary
$INCLUDE /etc/freeradius/dictionary.kamailio
```

Configuration du client freeRADIUS (Kamailio)

- Ajouter le dictionnaire kamailio dans le fichier /etc/radiusclient-ng/ dictionary

```
# The filename given here should be an absolute path.

$INCLUDE /etc/radiusclient-ng/dictionary.radius
```

- Ajouter l'adresse du serveur FreeRADIUS dans le fichier /etc/radiusclient-ng/servers

```
# RADIUS server to use for accouting requests. All that I
    # said for authserver applies, too.
     Acctserver          193.*.*.*
```

- Ajouter les lignes suivantes dans le ficher de configuration de kamailio (kamailio.cfg)

```
modparam("acc", "radius_config", "/etc/radiusclient-ng/radiusclient.conf")
modparam("acc", "radius_flag", 2)
modparam("acc", "radius_missed_flag", 3)
modparam("acc", "radius_extra", "User-Name=$Au; \
                        Calling-Station-Id=$from; \
                        Called-Station-Id=$to; \
                        Sip-Translated-Request-URI=$ruri; \
                        Sip-RPid=$avp(s:rpid); \
                        Source-IP=$si; \
                        Source-Port=$sp; \
                        Canonical-URI=$avp(s:can_uri); \
                        Billing-Party=$avp(s:billing_party); \
                        Divert-Reason=$avp(s:divert_reason); \
                        X-RTP-Stat=$hdr(X-RTP-Stat); \
                        Contact=$hdr(contact); \
                        Event=$hdr(event); \
                        ENUM-TLD=$avp(s:enum_tld)")
```

ANNEXE – 4 : Configuration module ACC avec MySQL

- Ajouter les lignes suivantes dans le ficher de configuration de kamailio (kamailio.cfg)

```
modparam("acc", "db_url", "mysql://root:root@193.*.*.*/accounting")

# flag to record to db
modparam("acc", "db_flag", 1)
modparam("acc", "db_missed_flag", 2)
# flag to log to syslog
modparam("acc", "log_flag", 1)
modparam("acc", "log_missed_flag", 2)
# use extra accounting to record caller and callee username/domain
# - take them from From URI and R-URI
modparam("acc", "log_extra",
"src_user=$fU;src_domain=$fd;dst_user=$rU;dst_domain=$rd")
modparam("acc", "db_extra",
"src_user=$fU;src_domain=$fd;dst_user=$rU;dst_domain=$rd")
```

ANNEXE – 5: Scripts IPtables

```
#####################################################
#
# Script  firewall start
#
#####################################################


# on rejette toutes les connections entrantes
iptables -t filter -P INPUT DROP

# on autorise la machine de supervision et accounting
iptables -A INPUT -s  193.*.*.* -j ACCEPT

# on autorise la machine d'administration
iptables -A INPUT -s  193.*.*.* -j ACCEPT


# on autorise les IPBX connectés au site pilote
# SITE1
iptables -t filter -A INPUT -s 193.*.*.* -p udp --sport 5060 -j ACCEPT

# SITE2
iptables -t filter -A INPUT -s 195.*.*.* -p udp --sport 5060 -j ACCEPT

echo "firewall start";
```

```
#####################################################
#
# Script stop firewall
#
#####################################################

iptables -t filter -P INPUT ACCEPT
echo  "firewall stop";
```